DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
ESCUELA SUPERIOR DE INGENIEROS
UNIVERSIDAD DE SEVILLA

Cooperative Perception Techniques for Multiple Unmanned Aerial Vehicles: Applications to the Cooperative Detection, Localization and Monitoring of Forest Fires

por

Luis Merino Cabañas

PROPUESTA DE TESIS DOCTORAL
PARA LA OBTENCIÓN DEL TÍTULO DE
DOCTOR INGENIERO DE TELECOMUNICACIÓN
SEVILLA, SEPTIEMBRE 2007

Director: **Prof. Dr.-Ing. Aníbal Ollero Baturone**

UNIVERSIDAD DE SEVILLA

Memoria presentada para optar al grado de Doctor Ingeniero de Telecomunicación por la Universidad de Sevilla

Autor: **Luis Merino Cabañas**

Título: **Cooperative Perception Techniques for Multiple Unmanned Aerial Vehicles: Applications to the Cooperative Detection, Localization and Monitoring of Forest Fires**

Departamento: **Departamento de Ingeniería de Sistemas y Automática**

V° B° Director

Aníbal Ollero Baturone

El autor:

Luis Merino Cabañas

A mis padres.
A mi sobrino Daniel.

Acknowledgments

There are many people with whom I am in debt for this thesis.

First of all, Professor Aníbal Ollero, director of this thesis and head of the Robotics, Vision and Control group, for giving me the opportunity for working in this amazing field.

I would like to thank Professors Günter Hommel and Domingos. X. Viegas for accepting to report this thesis. I should also thank Günter Hommel, Gösta Granlund, Raja Chatila and Simon Lacroix, for allowing me to work at their labs for some months along the thesis.

The partners of the COMETS project: Jérémi, Volker, Fran, Iker, Carsteen, Johan, Marek, for all that I have learnt from them, and for all the nice moments we had.

My colleges, and over all friends, Iván, Fernando, Ángel, Manuel, Joaquín, Antidio, Alberto, Manfredi, for their support, inspiration, never-ending discussions, laughs. Specially I should thank Iván for all these years of support and friendship.

I cannot forget the help of many past members of the Robotics and Vision group: Paco, Marcos, Víctor Blanco, Víctor Jara, Amanda, Íñigo.

My friends and colleges at Pablo de Olavide University: Virginia, Silvia, Carmen, Feli, Marisa, Begoña, for their support to the engineer geek.

María, Elena, Edu, Berta, María J., my friends, who have suffered all this thesis. And of course, all my friends in my hometown.

Reyes, for the good and bad moments along these last months. Moira, Mercedes, Antonio, thanks for your friendship.

I specially thank to my family, for all the comprehension and love along these years.

This thesis has been partially supported by the Spanish Ministry of Education (FPU-AP2000-3769).

Resumen

La robótica aérea tiene un gran potencial para la realización de tareas como la adquisición de datos e imágenes en áreas inaccesibles por medios terrestres, seguimiento, construcción de mapas, etc. Por otro lado, la complejidad de ciertas aplicaciones requiere la cooperación entre varios robots, debido a la necesidad de intervenir simultáneamente en distintas localizaciones, a la extensión espacial de la tarea a realizar o a las limitaciones de carga de los robots. Incluso en casos en los que la cooperación no es estrictamente necesaria, ésta puede ser empleada para incrementar la robustez de aplicaciones como detección y localización.

Esta tesis estudia el problema de la cooperación multi-robot para la percepción o *percepción cooperativa*. La percepción cooperativa se puede definir como la colaboración en un equipo de robots para la estimación del entorno, compartiendo información e incluso desarrollando acciones cooperativas. Más concretamente, la tesis considera la cooperación de múltiples vehículos aéreos no tripulados (UAVs en sus siglas inglesas) heterogéneos para la detección, localización, seguimiento y monitorización.

La percepción cooperativa comprende los problemas de la fusión de información y de la toma de decisiones cooperativas. Se analiza el problema de la percepción cooperativa bajo representaciones probabilísticas. En la tesis, se estudia la extensión de los filtros probabilísticos para la percepción multi-robot y se identifican los puntos principales a considerar. A continuación, estas ideas se aplican a diferentes representaciones y filtros. Concretamente, se describen algoritmos que extienden los filtros gausianos, filtros basados en matrices de celdas y filtros de partículas para la percepción multi-robot descentralizada.

La aplicación de la robótica aérea a la lucha contra los incendios forestales es una de las guías de la tesis, y, por tanto, las técnicas de percepción cooperativa desarrolladas se aplican al problema de la detección y localización automáticas de incendios forestales. Además, la tesis presenta un sistema para la monitorización de incendios, que permite incluir información de cámaras en el terreno y cámaras a bordo de robots y vehículos aéreos para determinar la posición y evolución del frente de llamas de un incendio.

Finalmente, la tesis presenta técnicas para la estimación de movimiento de un UAV empleando

visión; dichas técnicas se pueden emplear para la navegación; además, se presenta una posible extensión de los métodos para la estimación de la posición relativa de varios UAVs.

Las tesis presenta resultados obtenidos en experimentos de campo que involucran a tres UAVs y quemas controladas. Estos experimentos se emplean para validar las técnicas en tareas como la detección, confirmación, localización y monitorización de incendios mediante múltiples robots aéreos.

Abstract

Aerial robotics can be very useful to perform complex tasks such as data and image acquisition of areas otherwise inaccessible using ground means, localization of targets, tracking, map building and others. On the other hand, the complexity of some applications requires the cooperation between several robots. Moreover, even if cooperation is not required, this cooperation can be used to increase the robustness in applications such as detection and localization.

This thesis deals with multi-robot cooperation for the perception, or *cooperative perception*. Cooperative perception can be defined as the collaboration between a fleet of robots for the estimation of the state of the environment, by sharing information or even by developing cooperative actions. More precisely, this thesis considers the cooperation of multiple heterogeneous Unmanned Aerial Vehicles (UAVs) for detection, localization, tracking and monitoring purposes.

Cooperative perception embraces the problems of information fusion and the development of actions. The thesis analyzes the problem of cooperative perception under probabilistic representations of the knowledge. The extension of probabilistic filters to multi-robot perception is analyzed, and the main issues identified. Then, the thesis applies these ideas to different probabilistic representations and filters usually employed in the community. More precisely, the thesis describes algorithms that extend Gaussian filters, evidence grids and particle filters for decentralized multi-robot perception.

The potential application of aerial robots for fire fighting is one of the guides of the thesis, and thus, the cooperative perception techniques are applied to the automatic detection and localization of forest fires. Also, the thesis describes a cooperative perception system for forest fire monitoring, which considers information provided by static cameras and cameras on board aerial robots. This system is able to compute the evolution of the fire front automatically.

The thesis finally considers the use of vision for the estimation of the relative motion of an UAV, which can be applied to navigation tasks; moreover, a potential extension of the method to multi-UAV relative position estimation is described.

The thesis presents results obtained in field experiments involving three UAVs and controlled forest fires. These experiments are used to validate the presented techniques in the tasks of detection, confirmation, localization and monitoring of forest fires by means of a fleet of several UAVs.

Index

List of Figures

Notation

Along the thesis, matrix and vector variables are denoted by **bold** letters. Moreover, the thesis employs probability theory. In general, for random variables, capital letters denote the random variable $\mathbf{X}$ itself, while lowercase letters denote a particular realization of that variable $\mathbf{x}$. Also, the particular value of a quantity at a given time t is represented by using a temporal subindex, like $\mathbf{X}_t$, while a superindex $\mathbf{X}^t$ indicates a trajectory, that is, the set of values up to time t (that is, $\mathbf{X}^t = \{\mathbf{X}_0, \mathbf{X}_1 \dots \mathbf{X}_t\}$).

Mathematical notation

Chapter 2

$\mathbf{X}_t$, $\mathbf{X}^t$	State at time t, and the trajectory of the state up to time t
$\mathbf{z}_t$, $\mathbf{z}^t$	The measurements at time t, and the set of measurements up to time t.
$\mathbf{z}_{j,t}$	The measurements obtained by robot j at time t.
$\mathbf{u}_t$, $\mathbf{u}^t$	The action carried out at time t, and the set of actions carried out up to time t.
$\mathbf{u}_{j,t}$	The action carried out by robot j at time t.
$p(\mathbf{X}_t)$	The probability distribution function (pdf) over the random variable $\mathbf{X}_t$
$bel(\mathbf{X}_t)$	The belief state at time t, that is, the conditional probability distribution $p(\mathbf{X}_t \vert \mathbf{z}^t, \mathbf{u}^t)$
$bel_i(\mathbf{X}_t)$	The belief state for UAV i
$bel_{ij}(\mathbf{X}_t)$	The belief state due to the common information shared by UAVs i and j

Chapter 3

$\mathbf{Q}_{i,t}$	Pose of UAV i at time t.
$\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	Multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
$\boldsymbol{\xi}_t^i, \boldsymbol{\Omega}_t^i$	Information vector and matrix that express the belief at instant t for UAV i
$\boldsymbol{\xi}^{t,i}, \boldsymbol{\Omega}^{t,i}$	Information vector and matrix that express the belief over the state trajectory up to instant t for UAV i
$\boldsymbol{\xi}^{ij}, \boldsymbol{\Omega}^{ij}$	Information vector and matrix that represent the common information shared by UAVs i and j

Chapter 4

$\mathbf{r}_k$	Position associated to cell k.
$l^i_{k,t}$	Estimated Bernouilli probability (in log-odds) for cell k at instant t for UAV i
$l^{ij}_{k,t}$	Common information shared by UAVs i and j for cell k at instant t.
$\mathcal{B}(\alpha, \beta)$	Beta distribution with parameters α and β.
$\alpha^i_{k,t}$, $\beta^i_{k,t}$	Parameters of the Beta distribution for cell k at instant t for UAV i
$\alpha^{ij}_{k,t}$, $\beta^{ij}_{k,t}$	Common information shared by UAVs i and j for cell k (Beta distribution)

Chapter 5

$\mathbf{x}^{(k)}_{i,t}$	Particle representing a particular value of the state for UAV i.
$\omega^{(k)}_{i,t}$	Weight associated to particle k and UAV i.
$\pi(\mathbf{X}^t \vert \mathbf{z}^t)$	Proposal or importance distribution.

Chapter 6

f_k	The probability of being the fire at cell k.
q_k	The probability that cell k is completely burnt.
h_t	Estimated maximum height of the fire.

Chapter 7

$\mathbf{H}_{0i}$, $\mathbf{h}_{0i}$	Homography that relates frame i and the reference frame, and the same homography in vector form.
$\mathbf{R}_{0i}$, $\mathbf{t}_{0i}$	Rotation and translation that relate frame i and the reference frame.
$\mathbf{n}_0$	Unitary normal vector of the reference plane expressed in the reference frame.

Acronyms

AIIP	Application Independent Image Processing	IF	Information Filter
BBCS	BlackBoard Communication System	IMU	Inertial Measurement Unit
CC	Control Centre	KF	Kalman Filter
CI	Covariance Intersection	KL	Kullback-Leibler (divergence)
DACLE	Detection and Alarm Confirmation, Localization and Evaluation service	LOP	Linear Opinion Pool
DBN	Dynamic Bayesian Network	MDP	Markov Decision Process
EMS	Event Monitoring System	MLE	Multi-Level Executive
FOV	Field of View	POMDP	Partially Observable Markov Decision Process
GMM	Gaussian Mixture Model	UAV	Unmanned Aerial Vehicle
GPS	Global Positioning System	UT	Unscented Transform

Imagination is more important than knowledge. For knowledge is limited, whereas imagination embraces the entire world, stimulating progress, giving birth to evolution

Albert Einstein

Chapter 1

Introduction

This thesis presents contributions in the field of cooperative perception within robot teams. More precisely, cooperative perception among several Unmanned Aerial Vehicles is addressed. The potential application of aerial robots for fire fighting is also considered, and thus, the application of these perception techniques to automatic forest fires detection, localization and monitoring is analyzed.

This chapter firstly presents the motivation and the main objectives of the research carried out. Then, the outline and main contributions of the thesis are presented. Finally, the framework in which the work has been developed is described.

1.1 Motivation and objectives

Unmanned Aerial Vehicles (UAVs) are self-propelled air vehicles that are either remotely controlled or capable of conducting autonomous operations. Since the first UAV flew (in 1917, in World War One), UAVs have been mainly used in military applications and, in general, for classified purposes. Nevertheless, it is clear that UAVs have a wide range of civil applications. The higher mobility and maneuverability of UAVs with respect to ground vehicles makes them a natural approach for tasks like information gathering or even the deployment of instrumentation. Aerial robotics can be very useful to perform complex tasks such as data and image acquisition of areas otherwise inaccessible using ground means, localization of targets, tracking, map building and others. In recent years, the technologies for autonomous aerial vehicles have experienced an important development. This has made the research on aerial autonomous systems affordable for universities and civilian research centers.

On the other hand, the complexity of some applications requires the cooperation between several robots. Moreover, even if cooperation is not required, it can be used to increase the robustness in applications such as detection and localization.

This thesis deals with cooperation for the perception, or *cooperative perception*. Cooperative perception could be defined as the collaboration between a fleet of robots for the estimation of the state of the environment, by sharing information or even by developing cooperative actions. More precisely, this thesis considers the cooperation of multiple heterogeneous UAVs for detection,

localization, tracking and monitoring purposes. The heterogeneity increases the complexity of the problem, but also provides several advantages for the application, such as the possibility to exploit the complementarities of different UAV platforms with different mobility attributes and also different sensor and perception capabilities. And thus, the thesis shows results obtained with UAVs that are heterogeneous in terms of the platforms (helicopters and airships are considered), the sensors that the vehicles carry on board (infrared and visual cameras, fire detectors) and also in the decisional capabilities (from teleoperated to fully autonomous vehicles).

The thesis seeks the design of a distributed system for UAV cooperative perception, and its test in close to operational conditions on actual experiments involving several UAVs. And, although the techniques developed are extendable to other applications, the algorithms have been tested for the detection and monitoring of forest fires using aerial vehicles. The motivation for this application is easy to understand in southern countries in Europe.

Hundreds of thousands of hectares are devastated by wildfires each year. Forest fires lead to the destruction of forest and the wildlife that inhabits them, and have a disastrous social, economic and environmental impact. Forest fire fighting involves extensive human resources, and is a very dangerous activity, which originates many casualties every year. In many cases, the lack of information about the dynamic evolution of fire plays an important role in these accidents. Forest-fire fighting is commonly based on estimations made by fire fighting experts from visual observations. These estimations are subject to a high number of errors such as smoke occluding the flames, human inaccuracy in the visual estimation and errors in the localization of the fire. More recently, new technologies have been applied to the field of fire fighting. However, these platforms still have different technological and practical problems for their use in operational conditions, such as low reliability, high costs and others.

UAVs can mix the better of some of these technologies. As aerial systems, they take advantages over the ground means in terms of coverage capabilities and accessibility (see Fig. 1.1). Also, they avoid the human risk inherent to manned aerial means. Moreover, the costs involved can be lower than using conventional airplanes or helicopters. One of the objectives of this thesis is to demonstrate the feasibility of UAVs for forest fire fighting activities.

Thus, the cooperative perception techniques presented are tested within a forest fire fighting scenario, which is described more precisely in Chapter 8. The objective is that a fleet of UAVs is able to detect potential fire alarms in a given area, to confirm them and to localize them precisely. The presence of a fleet of heterogeneous vehicles is a key point for confirmation purposes, for the use of different sensors allows to reduce the false alarm ratio. Also, once a fire is detected, it is very interesting to obtain information about the dynamic evolution of the fire. The objective is that the fleet of robots monitors the fire, providing in real-time information about the fire front position and its evolution, the rate of spread of the fire, the height of the flames and other important parameters for fire fighting.

1.1.1 Objectives

Therefore, the main objectives of the research described in this thesis are the following:

Figure 1.1: UAVs can be very helpful in forest fire fighting activities. Here, two images of a controlled fire taken from an helicopter are shown.

- To define a framework for cooperative perception within a robot team. This framework must consider uncertainty in perception and actuation, information fusion among multiple robots and coordinated actions. The aim is the search for *efficient* and *decentralized* algorithms.

- The application of the previous framework to the task of cooperative detection, localization and tracking of objects of interest by a fleet of UAVs. Different approaches within the same framework will be compared.

- As a testbed, the results are applied to the cooperative detection of forest fire alarms. The objective is that a fleet of UAVs should be able to, cooperatively, determine the precise position of the potential fire alarms within a given area, while discarding false alarms.

- To apply cooperative perception techniques to the task of forest fire monitoring employing autonomous helicopters. In this case, the objective is to estimate the dynamic evolution of a forest fire, mainly the geographic position of the fire front and estimations of the height of the flames.

- The fusion of information requires to anchor all the data gathered by the team of UAVs to the same reference frame. Although the localization problem is not addressed specifically in this thesis, the objective is to develop methods based on vision for relative position estimation within a fleet of UAVs.

- To demonstrate the previous ideas in a experimental setup involving several UAVs, and to show the feasibility of the use of UAVs in fire fighting activities.

1.2 Outline and main contributions

The thesis consists of nine chapters complemented with three appendixes. A summary of the contents of the chapters is presented here:

In **Chapter 2**, the problem of cooperative perception in a multi-robot system is considered. A general formulation based on probability theory and Bayesian inference is presented. The state of the art on cooperative perception is reviewed, including a description of the related work in the field of aerial robotics. Part of the material presented in this chapter has been published in the journal *Annual Reviews in Control* (Ollero and Merino, 2004).

In **Chapter 3**, an algorithm for decentralized estimation based on the Information Filter is described. Then, it is applied to the problem of cooperative object detection, localization and tracking. Vision is used as the main sensor in this chapter. The likelihood functions for vision and other representation issues are discussed. Part of the material of this chapter led to the publications (Merino et al., 2005) (in the *International Conference on Robotics and Automation*, ICRA) and (Merino et al., 2006a) (in the *Journal of Field Robotics*).

Chapter 4 presents decentralized perception algorithms based on a common non-parametric approach for uncertain information representation: certainty grids. The use of Beta distributions for evidence grids is also considered. This representation is more suitable than that described in Chapter 3 in some applications. Part of this chapter has been published in (Merino et al., 2006b). Moreover, some of the material of the Chapters 3 and 4 has been published as a chapter (Merino et al., 2007a) of the book *Multiple Heterogeneous Unmanned Aerial Vehicles* (Ollero and Maza, 2007).

Particle filters have become a widespread tool for state estimation in the non-linear non-gaussian case. **Chapter 5** considers the use of particle representations for cooperative perception. The chapter presents some approaches for the decentralization of the particle filters in the multi-robot case, and its application to the detection and localization of objects.

The techniques described in previous chapters are applied to the case of event detection and localization. **Chapter 6** discusses the application of robotic techniques to forest fires monitoring. Fire monitoring consists on the estimation of the dynamic evolution of a forest fire, including the computation of the fire front spread, height of the flames and other parameters. This chapter presents a system for fire monitoring using cameras. Results from actual forest fires experiments are presented. The description of these experiments and part of the results with static cameras were published in (Viegas et al., 2002). Also, part of the chapter has been accepted for publication in the journal *Image and Vision Computing* (Martínez de Dios et al., 2007a).

The ideas are extended in order to build a fire monitoring application using several UAVs. Chapter 6 then extends the methods described to cope with moving cameras on board aerial robots. Experimental results are also presented. Initial results and ideas on the use of UAVs for forest fire monitoring were published in (Merino and Ollero, 2002b), in the *International Conference on Intelligent Robots and Systems (IROS)*, and (Merino and Ollero, 2002a). More recently, part of the chapter led to the publications (Merino et al., 2004; Martínez de Dios et al., 2006b).

Chapter 7 deals with an important problem that is assumed solved in the previous chapters: UAV motion estimation and relative localization. The chapter presents a visual odometer for UAV motion estimation. The application of this technique for UAV pose estimation, landing and building inspection is considered. Then, the techniques are also applied to multi-UAV relative position

estimation. The first part of the chapter has been published in (Caballero et al., 2005; Caballero et al., 2006), in ICRA. Also, the chapter has lead to a publication in the *IEEE Robotics and Automation Magazine* (Merino et al., 2006c) and also as a chapter (Merino et al., 2007b) of the book *Advances in Unmanned Aerial Vehicles. State of the art and the road to autonomy* (Valavanis, 2007).

Finally, **Chapter 8** describes the general architecture of the multi-robot team in which the techniques have been tested. The chapter describes the distributed cooperative perception system implemented as a result of the techniques presented in this thesis. Part of the chapter has been published in the *IEEE Robotics and Automation Magazine* (Ollero et al., 2005). Moreover, the chapter describes the demonstration scenario in which the previously presented techniques and systems have been tested, and presents results on actual experiments of fire detection, localization and monitoring, involving controlled fires. These results and experiments have been detailed in the publications (Merino et al., 2006a; Martínez de Dios et al., 2007b).

The thesis is completed with **Chapter 9**, which discusses and concludes the results of the thesis and in which the future work is summarized.

1.3 Framework

The main part of this work has been performed in the frame of the European project **COMETS** (Real-time Coordination and cOntrol of Multiple hETerogeneous unmanned aerial VehicleS, IST-2001-34304). The COMETS project lasted from May 2002 till July 2005. This project is probably the first research project on multiple autonomous aerial vehicles for non-military applications in Europe.

The main objective of COMETS[1] was to design and implement a distributed control system for cooperative detection and monitoring using heterogeneous UAVs. In order to achieve this general objective, the project considered the following subgoals:

- The design and implementation of a new control architecture for the operation of multiple cooperative heterogeneous aerial systems including both autonomous and teleoperated systems. The developed architecture allows for several multi UAV operation schemes including central planning and control, simple coordination, and cooperative task achievement.
- The development of new control techniques including optimal coordination and control of the multi UAV team; the analysis of the stability of helicopter flight and the development of model-based fault detection methods.
- The integration of distributed sensing techniques and real-time image processing capabilities that are necessary to achieve the previous objectives and to support the mission execution.

The work presented in this thesis is mainly related with the third and first subgoals of the COMETS project.

[1] http://grvc.us.es/comets

Figure 1.2: Images from the Lousã airfield in which some of the field experiments were carried out.

In order to test and validate these concepts and technologies, the system developed has been demonstrated in forest fire detection, confirmation, localization and monitoring. The research on forest fire behavior and prediction is the main objective of the **SPREAD** European project (Forest fire spread prevention and mitigation). SPREAD aimed to the development and implementation of an integrated forest fire management system for Europe, with inputs from Earth observation and meteorological data, information on the human dimension of fire risk, and assimilation of these data in fire prevention and fire behavior models. This project offered the framework for the development of techniques for fire monitoring using aerial vehicles, which are also described in this thesis.

Both projects had an strong experimental emphasis. The results of this thesis have been obtained in the experiments performed at Lousã and Gestosa, Portugal (see Figs. 1.1 and 1.2), as part of the previous projects . They offered an unique possibility to test robotic systems in close to operational conditions. In the Lousa experiments, the UAVs Marvin, Karma, and Heliv, which will be described in Chapter 8, were involved.

Besides, the thesis has been developed also within the following projects, that have provided funding and equipment needed for the research:

- **AEROSENS** (AErial RObots and SENSor networks with mobile nodes for cooperative perception) Funded by the Dirección General de Investigación. DPI2005-02293. 2005-2008. The project aims at the development of a system based on the use of aerial and ground robots and sensor networks for cooperative perception. The system is based on the joint application of Aerial Robotics and the technology of Wireless Sensor Networks.

- **AWARE** (platform for Autonomous self-deploying and operation of Wireless sensor-actuator networks cooperating with AeRial objEcts). Funded by the European Commission (IST2006-33579). The general objective of the project is the design, development and experimentation of a platform providing the middleware and the functionalities required for the cooperation among aerial flying objects, i.e. autonomous helicopters, and a ground sensor-actuator wireless network, including mobile nodes carried by people and vehicles.

- **URUS** (Ubiquitous networking Robotics in Urban Settings). Funded by the European Commission (IST-045062). The URUS project focuses on designing and developing a network of robots that, in a cooperative way, interacts with human beings and the environment for tasks

of guidance and assistance, transportation of goods, and surveillance in urban areas. Specifically, the objective is to design and develop a cognitive networked robot architecture that integrates cooperating urban robots, intelligent sensors (video cameras, acoustic sensors, etc.), intelligent devices (PDA, mobile telephones, etc.) and communications.

It is worth to say that part of the thesis has been developed during several stays in Universities abroad, at the Computer Vision Laboratory[2] of the Linköping University, at Linköping, Sweden (August-November 2001), the Real-time Systems and Robotics Group[3] of the Technical University of Berlin (August-November 2002) and the Robotics and Artificial Intelligence Group[4] of the Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) at Toulouse (September 2004). All these centers are currently involved (or have been involved) in the development of aerial robots.

[2]http://www.cvl.isy.liu.se/
[3]http://pdv.cs.tu-berlin.de/MARVIN/index.html
[4]http://www.laas.fr/~simon/eden/

If I have seen further it is by
standing on the shoulders of Giants
Sir Isaac Newton
Letter to Robert Hooke, 1676

Chapter 2

Cooperative perception in multi-robot systems

This chapter deals with the cooperative perception problem in multi-robot systems. A discussion on its main issues and some important definitions for the main topics of this thesis are presented. Moreover, the related work on multi-robot perception is summarized. Finally, the special characteristics and state of the art in UAV research is covered, including multi-UAV systems. In the following chapters, more specific related work to each problem is presented where necessary.

2.1 Introduction

Perception systems are employed by a robot to estimate the static and dynamic characteristics of the environment (for instance, the position of the obstacles that surround the robot or the position of a vehicle being tracked) and the own robot state. This estimation is obtained from the data gathered by the different sensors on board the robot. For instance, a robot could use a 2D laser range-finder to determine the obstacles around, a camera to track moving persons, etc.

In the case of the team of robots, one can do better than a robot perceiving alone. The information that each robot obtains about the environment can be *shared* to improve the perception, so that each robot obtains a better picture of the world than if it was alone (Fig. 2.1). Moreover, from the perception point of view, the robots can explicitly *cooperate*, developing actions to collect data. Thus, *cooperative robot perception* could be defined as the collaboration inside a fleet of robots for the estimation of the state of the environment, by sharing information or even by developing cooperative actions.

2.1.1 General issues

One of the main objectives of the thesis is to develop a cooperative perception system for a multi-UAV fleet. When considering multi-robot perception, there are several important issues that should be addressed.

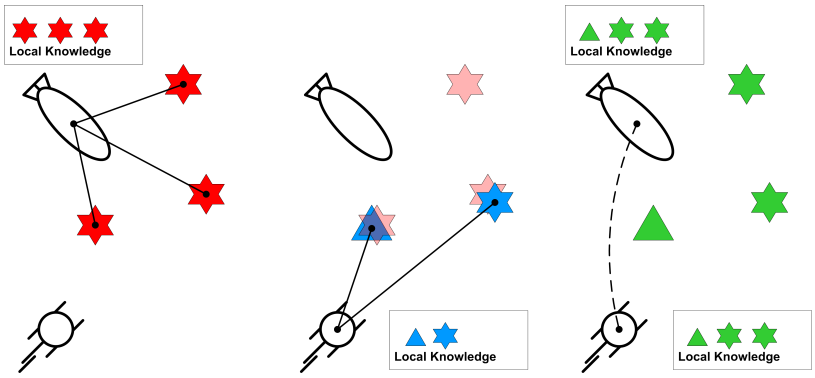

Figure 2.1: Each robot alone only perceives part of the environment (left and centre). Information sharing (right) allows a robot to obtain information about part of the environment not covered by its sensors (alarm up right for the helicopter), to improve the information about known objects (for instance, to refine the position of the alarm in the middle), or even to improve the identification of objects surrounding (for instance, to discard false alarms).

Knowledge representation and information fusion

Robots use their sensors to update their knowledge. This local knowledge is usually arranged into a hierarchy from lower to higher abstraction levels, ranging, for instance, from the raw sensor readings, estimations of the own robot position and velocity or the position of the surrounding obstacles to the estimation of the shape and appearance aspects of an object and even the identity of a particular object.

If a robot in a fleet can communicate with its companions, then the information received from other robots can be fused with the local one to improve this knowledge. There can be an improvement due to new information, to parts of the world not previously known. Moreover, if uncertainty is considered, this improvement can consist of reducing the uncertainty associated to a part of the world that is observed by different teammates. Information fusion is, then, a key issue in a cooperative perception system. In the most general case, the information to be fused can correspond to different levels of abstraction, as commented before.

Information fusion requires to translate the received information to the robot's local representation; therefore, in general, the fusion architecture will influence the local representation employed by each robot. Also, the problem of data association is an important aspect that should be solved in order to determine if two robots are referring to the same part of the world.

Information fusion rules should lead to an improved knowledge about the world, avoiding inconsistent estimates. As it will be seen, special care should be taken to avoid *rumor propagation* in the case of decentralized systems, which can lead to overoptimistic estimations.

Cooperation

Robots use locally the perception estimates to react under a changing world and even to develop plans if they are endowed with decisional abilities. These actions can include tasks that improve the

Table 2.1: Taxonomic axes by (Dudek et al., 2002)

Axis	**Description**
Collective size	The number of robots
Communication range	The maximum distance at which two robots can communicate
Communication topology	Of the robots within communication range, those which can be communicated with
Communication bandwidth	Amount of information that can be transmitted
Collective reconfigurability	The rate at which the organization of the fleet can be modified
Processing ability	Computational model of each individual
Collective composition	Whether the robots are homogeneous or heterogeneous

local knowledge of the world, like visiting unknown zones or moving to better points of view.

In the case of multi-robot perception, robots can explicitly coordinate themselves in information gathering tasks. Even, there could be certain perception tasks that only can be accomplished if a fleet of physically separated robots is used (for instance, observing an event from different vintage points).

All in all, it is required to define *metrics* about the information gain that the actions of a particular robot or a group of robots in the fleet produce from the point of view of perception.

Other issues

Apart from the information fusion and cooperation issues, there are other important aspects that should be considered as well. A robot team can be also seen as a set of complex sensors that provide information about the environment. One issue is whether this information is combined in a purely centralized fashion (all the data are collected in a central node that builds a world representation) or in a decentralized manner, in which each robot compounds its own model and communication occurs at a higher level. Also, the architecture considered can be something in between these two extremes. This will influence when the robots should communicate, what should be communicated, and others. In general, centralized solutions can provide optimal solutions, but they do not scale well with the number of robots and are much prone to communication failures.

Of course, the solution will depend on several aspects, like the physical communication layer available, the local processing power of the robots, the tasks to be accomplished, the structure of the environment, the autonomy with which the individual robots should be endowed, etc. In order to guide the answer to some of these questions, the next section will characterize the nature of the particular multi-robot system considered.

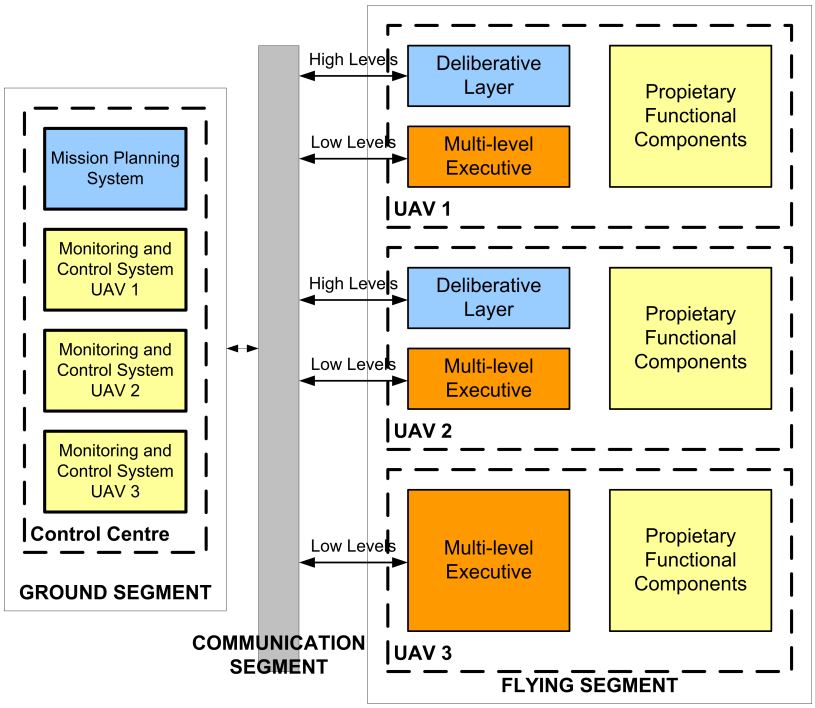

Figure 2.2: General architecture of the COMETS system. The different robots of the flying segment are heterogeneous even in terms of decisional capabilities, ranging from teleoperated systems to fully autonomous vehicles. In the lower levels, the robots can synchronize the execution of tasks. In the higher levels the decisional layers collaborate in the tasks planning and allocation procedures, as well as in coordination and cooperation. A Control Centre is in charge of generating general plans and monitor the status of the fleet.

2.1.2 Limiting the scope: the multi-robot architecture

The perception architecture that will be presented through this thesis has been developed in the frame of the COMETS project. As commented before, the system developed in this project pursues the coordination and control of a fleet of heterogeneous UAVs (ranging from helicopters to blimps, from fully autonomous to teleoperated vehicles). The details of the fleet are given in Chapter 8. Considering the taxonomy presented by (Dudek et al., 2002) (see Table 2.1), the COMETS multi-robot system can be cataloged as (Gancet, 2005):

SITE-LIM The number of robots considered is limited, although the number could be in the range of 10 robots.

COM-NEAR The robots can only communicate with other robots within a certain range

TOP-GRAPH The communication topology can be quite general

BAND-INF The cost of communication is negligible compared to other costs.

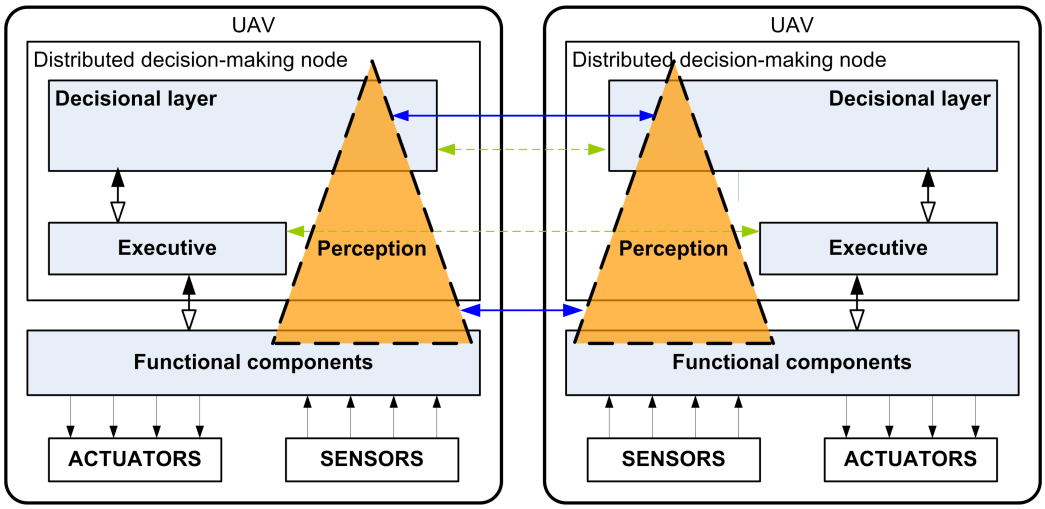

Figure 2.3: Single robot architecture and relations. Each robot can be modeled as following and hybrid deliberative-reactive architecture. The functional components allow the UAV to accomplish a set of simple atomic tasks at requests from the executive (which is also responsible of task synchronization with other UAVs). The decisional layer deals with the problem of multi-robot task planning and allocation. The perception modules provide the required information.

ARR-DYN The relationship among robots can be reconfigured dynamically.

PROC-TME Regarding the processing ability, each robot can be thought as Turing Machine Equivalent.

CMP-HET The robots are highly heterogeneous.

In order to coordinate a fleet of robots of these characteristics, a control architecture was developed, and is the main topic of the thesis of Gancet (2005). This architecture can deal with UAVs of different levels of autonomy (Gancet and Lacroix, 2004; Gancet et al., 2005). Although more details are given in Chapter 8, Fig. 2.2 shows a very schematic architecture of the actual testing.

In the highest level of autonomy considered, each robot can be modeled as following an hybrid deliberative-reactive architecture, similar to that described in (Alami et al., 1998a), see Fig. 2.3. The functional components correspond to proprietary modules of each UAV in charge of generating the control loops needed to accomplish elementary or atomic tasks, like taking off, landing, flying waypoints or information gathering tasks. The executive is in charge of the UAV supervision and elementary task control, and also communicates with the executive of other UAVs for tasks synchronization. The deliberative layer deals with local planning and supervision, as well as with high level coordination, task planning and task allocation in the multi-UAV fleet (Gancet et al., 2005).

The perception system lies parallel to the previous decisional architecture. In each layer, the perception modules should provide the needed information about the environment (whether for navigation purposes in the more reactive functional layers or for high level planning). Moreover, the control architecture allows for tasks coordination among robots by explicit communication of tasks

status, synchronization signals and others. In the same way, the perception architecture should allow the robots to share knowledge and information obtained from the local perception systems.

Given the characteristics of the fleet and the missions that they will face (see Chapters 1 and 8), it is quite likely that, during operation, communication failures will occur due to robots going out of communication range or droppings of the available bandwidth. Also, the topology of the communication channels can change dynamically. All these issues suggest a loose cooperation in perception, that is, a decentralized perception scheme in which each robot locally processes its data and builds its own world model (which will be required for their autonomous operation). Then, the robots will share their local estimations about the environment if they can communicate. Although there is a control centre for monitoring and control, it will be assumed that there is no central node in charge of information fusion.

Hence, firstly the techniques for local perception and information representation are described in the next section. This will lead to a discussion on the communication and information fusion issues that can be derived from the local schemes.

2.2 Robot local perception

Perception can be seen as an inference problem, in which, from the data gathered by their sensors and the actions performed, the robot infers the state of the environment (Russell and Norvig, 2003). Sensors are inherently *uncertain*. But not only sensors. Uncertainty comes also from nearly every aspect related to robotics. (Russell and Norvig, 2003; Thrun et al., 2005). Thus, models, control actions, the environment itself and indeed the algorithms running on a real robot give rise to uncertainty. Potential sources of uncertainty are random noise, vagueness, ignorance, model inaccuracies and other aspects.

It is generally accepted that any system capable of uncertain reasoning is generally more robust than one that does not. Then, the local perception algorithms should be able to handle uncertain data, and to maintain information about this uncertainty. At most, what a robot can obtain is a *degree of belief*, that is, an estimation of the true state and the uncertainty associated to that estimation.

The framework that will be used for uncertain reasoning is probability theory. In the last decade, probabilistic algorithms for uncertain reasoning have become one of the most prominent tools among roboticists. Of course, there are other options that will be discussed at the end of the chapter.

In probabilistic perception, the current knowledge is represented by a probability distribution, that assigns to each possible value of the state space a probability, and that is called the *belief state*. At time t, this probability distribution is the conditional probability distribution:

$$bel(\mathbf{X}_t) = p(\mathbf{X}_t|\mathbf{z}^t, \mathbf{u}^t). \tag{2.1}$$

that is, the probability distribution on the state $\mathbf{X}_t$ at time t conditioned on all the information gathered up to time t, $\mathbf{z}^t$, and the actions carried out, $\mathbf{u}^t$ [1]. From a probabilistic point of view,

[1] note that lower case letters ($\mathbf{z}^t$ and $\mathbf{u}^t$) are used to represent actual realizations of random variables, which are

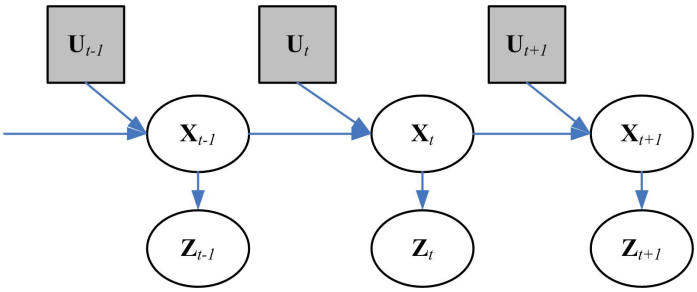

Figure 2.4: Dynamic Bayes Network that summarizes the relations between state, actions and measurements.

this conditional distribution, called the *posterior*, represents all the information that a robot can compute from the sensor data collected, and is thus the internal information representation that will be considered. This posterior takes into account the uncertainties due to data gathering and control, model inaccuracies, noise, etc. Once the previous function is known, some other measurements can be obtained (such as the mean or other moments, etc).

Computing (2.1) at each time t is known as the filtering problem (Stone et al., 1999). Sometimes it is also needed to know the belief over the trajectories of the state up to time t, represented by the distribution $p(\mathbf{X}^t|\mathbf{z}^t, \mathbf{u}^t)$.

As it will be seen, computing the conditional probability in (2.1) requires to know how the different variables are related; the most important relations to be considered are independence and conditional independence. One way to represent these relations are the so-called Dynamic Bayesian Networks (DBNs) (Kjaerulff, 1992). Figure 2.4 shows the typical network structure for relations between the state, the data gathered and the robot actions.

The details are presented below, but what the network states is that the actions of the robot influence the state, which also depends only on the state at time $t-1$ (the Markov assumption). On the other hand, the data obtained at time t only is conditionally independent of the other variables given the state of the environment and the robot at that time (the complete state assumption).

2.2.1 Bayesian inference and Bayes rule

The main framework for probabilistic state estimation when new information is available is Bayesian inference (Stone et al., 1999; Thrun et al., 2005). Bayesian inference allows to integrate measurements gathered by the sensors with prior knowledge about the state to obtain an updated estimation of the belief on the state.

A central theorem in Bayesian inference is the well-known Bayes rule. The Bayes rule allows to relate conditional probability distributions. If X and Y are random variables, and y is the known value of Y, then the Bayes rule allows to obtain the probability distribution on X conditioned on $Y = y$.

represented by upper case letters ($\mathbf{Z}^t$ and $\mathbf{U}^t$)

$$p(X|y) = \frac{p(y|X)p(X)}{p(y)} = \frac{p(y|X)p(X)}{\int p(y|X)p(X)dX} \tag{2.2}$$

Given an initial knowledge on a random quantity X, $p(X)$ (the *prior* distribution), and new data (y), the Bayes rule allows to update the knowledge on X after incorporating these data, obtaining $p(X|y)$ (the *posterior* distribution on X). A central role is played by the *likelihood function* $p(y|X)$, which states the probability of obtaining data y given the values of the quantity X.

2.2.2 Bayes filters for robot local perception

A robot acting in the real world should update its belief state each time it receives new data gathered by its sensors. It is important that the belief state is estimated recursively. Recursive estimation implies that the belief state at time t, $p(\mathbf{X}_t|\mathbf{z}^t, \mathbf{u}^t)$ should be computed from the belief state on a previous time instant $p(\mathbf{X}_{t-1}|\mathbf{z}^{t-1}, \mathbf{u}^{t-1})$, considering the actions taken $\mathbf{u}_t$ and after incorporating the new information $\mathbf{z}_t$.

Using the Bayes rule (2.2), the posterior (2.1) can be expressed as:

$$p(\mathbf{X}_t|\mathbf{z}^t, \mathbf{u}^t) = \frac{p(\mathbf{z}_t|\mathbf{X}_t, \mathbf{z}^{t-1}, \mathbf{u}^t)p(\mathbf{X}_t|\mathbf{z}^{t-1}, \mathbf{u}^t)}{p(\mathbf{z}_t|\mathbf{z}^{t-1}, \mathbf{u}^t)} \tag{2.3}$$

The normalizing term $p(\mathbf{z}_t|\mathbf{z}^{t-1}, \mathbf{u}^t)$ is an scalar independent of $\mathbf{X}_t$ that will be denoted by $1/\eta$.

Two assumptions allow to obtain a recursive estimation of the state. First, the stochastic process associated to the state $\mathbf{X}(t)$ should be a first-order Markov process (Jazwinski, 1970; Stone et al., 1999). That is, if the state is known at time t, the future is independent of the past (i.e., $p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{X}_{t-2}, \ldots) = p(\mathbf{X}_t|\mathbf{X}_{t-1})$). The second assumption is that the information about the state at time t is enough to determine the measurement at time t, $\mathbf{z}_t$ (that is, measurements at different time instants are independent, given the state). Both assumptions are summarized in the conditional independence relations of the network of Fig. 2.4.

The second assumption allows to simplify the term:

$$p(\mathbf{z}_t|\mathbf{X}_t, \mathbf{z}^{t-1}, \mathbf{u}^t) = p(\mathbf{z}_t|\mathbf{X}_t). \tag{2.4}$$

To relate the current state to the previous estimation, the second term in the numerator can be expanded using the theorem of total probability as:

$$p(\mathbf{X}_t|\mathbf{z}^{t-1}, \mathbf{u}^t) = \int p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{z}^{t-1}, \mathbf{u}^t)p(\mathbf{X}_{t-1}|\mathbf{z}^{t-1}, \mathbf{u}^t)d\mathbf{X}_{t-1} \tag{2.5}$$

As the state is a first-order Markov process, then,

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{z}^{t-1}, \mathbf{u}^t) = p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t) \tag{2.6}$$

Using (2.4), (2.5) and (2.6), (2.3) becomes

$$p(\mathbf{X}_t|\mathbf{z}^t, \mathbf{u}^t) = \underbrace{\eta p(\mathbf{z}_t|\mathbf{X}_t)}_{update} \underbrace{\int p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t)p(\mathbf{X}_{t-1}|\mathbf{z}^{t-1}, \mathbf{u}^{t-1})d\mathbf{X}_{t-1}}_{prediction} \tag{2.7}$$

Equation (2.7) is the discrete time recursion that allows to update the belief state after a robot has performed an action $\mathbf{u}_t$ and received new data $\mathbf{z}_t$. It is usually decomposed in two steps, called *prediction* and *update*. Prediction usually implies an increase in the amount of uncertainty on $\mathbf{X}$, while updating narrows this uncertainty as a consequence of the new measurement.

The important terms in the previous expressions are the conditional distributions $p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t)$ (the *transition* model) and the *likelihood* $p(\mathbf{z}_t|\mathbf{X}_t)$ (also called the *measurement* model) and the conditional dependencies among the random variables.

Using the same assumptions, it is possible to derive the Bayes filter for the recursive estimation of the belief for the full state trajectory $p(\mathbf{X}^t|\mathbf{z}^t, \mathbf{u}^t)$ (Stone et al., 1999):

$$\begin{aligned} p(\mathbf{X}^t|\mathbf{z}^t, \mathbf{u}^t) &\overset{Bayes}{=} \eta p(\mathbf{z}_t|\mathbf{X}^t, \mathbf{z}^{t-1}, \mathbf{u}^t) p(\mathbf{X}^t|\mathbf{z}^{t-1}, \mathbf{u}^t) \\ &= \eta p(\mathbf{z}_t|\mathbf{X}_t) p(\mathbf{X}_t|\mathbf{X}^{t-1}, \mathbf{z}^{t-1}, \mathbf{u}^t) p(\mathbf{X}^{t-1}|\mathbf{z}^{t-1}, \mathbf{u}^t) \\ &\overset{Markov}{=} \eta p(\mathbf{z}_t|\mathbf{X}_t) p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t) p(\mathbf{X}^{t-1}|\mathbf{z}^{t-1}, \mathbf{u}^{t-1}) \end{aligned} \tag{2.8}$$

Another convenient expression for this equation, that unrolls the previous expression until considering the *prior* information $p(\mathbf{X}_0)$, is:

$$p(\mathbf{X}^t|\mathbf{z}^t, \mathbf{u}^t) = \eta^{'} p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} p(\mathbf{z}_\tau|\mathbf{X}_\tau) p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1}, \mathbf{u}_\tau) \tag{2.9}$$

2.3 Bayesian multi-robot perception

In the case of a multi-robot fleet the objective is to cooperatively estimate the state of the world (that is, the relevant information, represented by $\mathbf{X}_t$) from the measurements and actions carried out by all the robots.

A decentralized approach has been proposed above, in which each robot shares its local estimations with its companions. In order to answer to the questions of what information should be communicated and how this information should be fused with the local knowledge of each robot, first it is analyzed what the resultant knowledge would be in the case that all information were available at any point of the fleet. That could be considered an ideal "omniscient" situation, in which a central node gets all the available information at any time.

Figure 2.5 represents the DBN for the multi-robot case (for two robots), under the same assumptions than before. The measurements will be the collection of all the measurements gathered by all the sensors of the M_t robots of the fleet $\mathbf{z}_t^m = [\mathbf{z}_{1,t}^T, \ldots, \mathbf{z}_{M_t,t}^T]^T$. Also, if $\mathbf{u}_t^m$ denotes the set of actions carried out by the robots, then, the belief state employing all the data of the fleet is given by:

$$bel_m(\mathbf{X}_t) = p(\mathbf{X}_t|\mathbf{z}^{m,t}, \mathbf{u}^{m,t}) = \eta p(\mathbf{z}_t^m|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t^m) p(\mathbf{X}_{t-1}|\mathbf{z}^{m,t-1}, \mathbf{u}^{m,t-1}) d\mathbf{X}_{t-1} \tag{2.10}$$

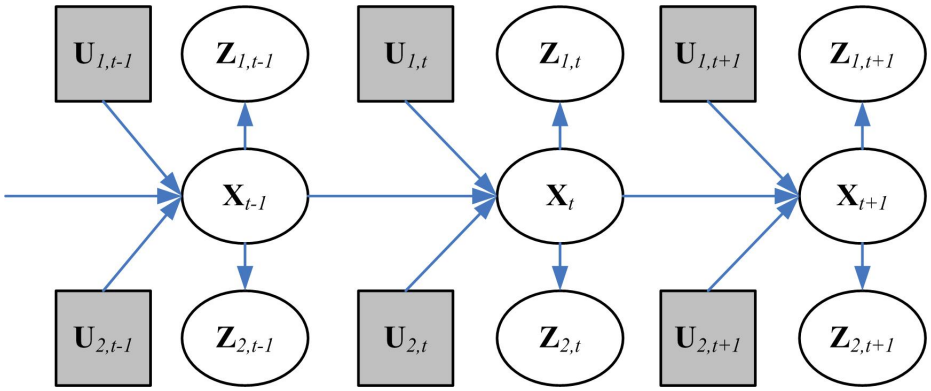

Figure 2.5: Bayes Network that shows the conditional independence relations for the case two robots.

The structure of the network in Fig. 2.5 arises from the assumption that the data gathered by the different robots at any time instant t are *conditionally independent* given the state $\mathbf{X}_t$, so the previous equation becomes:

$$bel_m(\mathbf{X}_t) = \eta \prod_{j=1}^{M_t} p(\mathbf{z}_{j,t}|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t^m) p(\mathbf{X}_{t-1}|\mathbf{z}^{m,t-1}, \mathbf{u}^{m,t-1}) d\mathbf{X}_{t-1} \tag{2.11}$$

For data fusion purposes, it will be very important to determine the likelihood function $p(\mathbf{z}_{j,t}|\mathbf{X}_t)$ for every sensor within the robots of the fleet.

Likewise, in the multi-robot case, equation (2.9) becomes:

$$bel_m(\mathbf{X}^t) = p(\mathbf{X}^t|\mathbf{z}^{m,t}, \mathbf{u}^{m,t}) = \eta^{'} p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} \Big[\prod_{j=1}^{M(\tau)} p(\mathbf{z}_{j,\tau}|\mathbf{X}_\tau) \Big] p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1}, \mathbf{u}_\tau^m) \tag{2.12}$$

The computation of Equations (2.11) and (2.12) are the ideal objective for the cooperative perceptions algorithms that will be presented in the next chapters of the thesis. The main assumption is that this central belief is the best that can be obtained from the data gathered. The decentralized fusion rules will try to obtain or at least to approximate this result.

Regardless of the method (if any) for the computation of the integrals on (2.11) or (2.12), one way to solve them is to communicate to a central node all the data and actions carried by the robots of a fleet, $\mathbf{z}_t^m, \mathbf{u}_t^m$, and try to infer the global belief state.

In theory, considering all the information in a central node would allow for an optimal computation of the belief state in the multi-robot case, in the sense that all the data are available when solving the previous equations. However, this solution implies to have enough bandwidth available to receive all the data, that can range from binary detections to full colour images. If a dynamic environment is considered, then the maximum delay allowed for data reception is bounded, should the belief be computed online.

Moreover, the measurement model $p(\mathbf{z}_{j,t}|\mathbf{X}_t)$ for all sources of data should be learnt and known in the centralized node, which therefore should maintain a database with these models. Besides, the

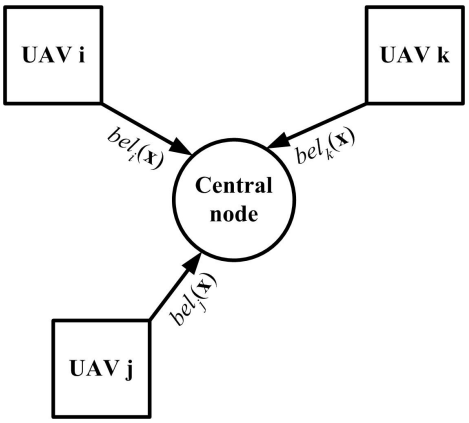

Figure 2.6: Scheme that shows the situation in the semi-decentralized scheme.

potential effects of all the actions that the robots can perform should be also modeled, in order to compute $p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t^m)$.

The bandwidth burden can be reduced by performing pre-processing functions locally, so each robot obtains interesting features from the raw data, and then sends only these features. Also, the robots can send directly the likelihood functions $p(\mathbf{z}_{j,t}|\mathbf{X}_t)$. However, this method does not scale well with the number of robots. Also, in general, it is not possible to dispose of all the data in a central node without delays due to bandwidth limitations, bounded communication ranges, etc, and this likelihood functions should be received without fails in order to obtain the ideal belief stated by eq. (2.11).

On the other hand, if the robots have decisional capabilities, then they should maintain their own local belief states that will be used by their decisional layers for planning activities and so forth. Therefore, the idea would be to combine in some way these local estimations, by communicating high-level belief states instead of raw data. Also, as it will be seen in the next section, the belief states summarize all the data gathered by the robot. Therefore, the communication of the belief states is more robust under communication failures.

The main question is if there is a way of combining the belief states so that the final belief state is closer (ideally the same) to the global belief that could be computed in the centralized case.

2.4 Semi-decentralized belief computation

In a decentralized approach, the robots *share* their beliefs with their neighbors. Then, the received information is locally fused in order to improve the local perception of the world. Before getting into the issues related to the fully decentralized computation, this section analyzes a first approach, depicted in Fig. 2.6. This approach will be named semi-decentralized. In it, each robot i maintains its local belief state $bel_i(\mathbf{X}_t)$ based on its own local data $\mathbf{z}_i^t$ and actions $\mathbf{u}_i^t$:

$$bel_i(\mathbf{X}_t) = p(\mathbf{X}_t|\mathbf{z}_i^t, \mathbf{u}_i^t) = \eta_i p(\mathbf{z}_{i,t}|\mathbf{X}_t) \int p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_{i,t}) p(\mathbf{X}_{t-1}|\mathbf{z}_i^{t-1}, \mathbf{u}_i^{t-1}) d\mathbf{X}_{t-1} \tag{2.13}$$

where $\eta_i^{-1} = p(\mathbf{z}_{i,t}|\mathbf{z}_i^{t-1}, \mathbf{u}_i^t)$. Then, the robot sends its belief state to a central node that combines all the local beliefs to obtain a global one. Firstly, potential fusion rules for this case are devised.

2.4.1 The case of static state

If the state is static, then it does not depend on the actions $\mathbf{u}^{m,t}$ and

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}) = \delta(\mathbf{X}_t - \mathbf{X}_{t-1}) \tag{2.14}$$

where $\delta(\mathbf{x})$ is the Dirac delta function. That is $\mathbf{X}_t = \mathbf{X}_{t-1} = \ldots = \mathbf{X}_0 = \mathbf{X}$. Thus, the belief state at time t using all robot data is given by:

$$\begin{aligned} bel_m(\mathbf{X}_t) = p(\mathbf{X}|\mathbf{z}^{m,t}, \mathbf{u}^{m,t}) = \eta p(\mathbf{z}_t^m|\mathbf{X}) bel_m(\mathbf{X}_{t-1}) = \\ = \eta^{'} p(\mathbf{z}_t^m|\mathbf{X}) p(\mathbf{z}_{t-1}^m|\mathbf{X}) bel_m(\mathbf{X}_{t-2}) = \ldots = \\ = \eta_m p(\mathbf{X}) \prod_{\tau=1}^{\tau=t} p(\mathbf{z}_\tau^m|\mathbf{X}) \end{aligned} \tag{2.15}$$

where the time index has been removed to indicate that the state is static, and then $bel(\mathbf{X}_t)$ means the belief state after all the data gathered up to time t has been integrated. Similarly, for any robot i:

$$bel_i(\mathbf{X}_t) = \eta_i p(\mathbf{X}) \prod_{\tau=1}^{\tau=t} p(\mathbf{z}_{i,\tau}|\mathbf{X}) \tag{2.16}$$

If M robots are considered, and given the conditional independence assumption $p(\mathbf{z}_t^m|\mathbf{X}) = \prod_{i=1}^{M} p(\mathbf{z}_{i,t}|\mathbf{X})$, then, if the prior beliefs $p(\mathbf{X})$ are the same:

$$bel_m(\mathbf{X}_t) = \eta p(\mathbf{X}) \prod_{i=1}^{M} \frac{bel_i(\mathbf{X}_t)}{p(\mathbf{X})} \tag{2.17}$$

This equation gives a basic formula to combine the robots beliefs in order to obtain the global one. It means that the central node directly combines all the beliefs received, after "removing" the common information that all robot share (the prior $p(\mathbf{X})$). Another convenient way of representing the previous relations is in recursive form:

$$bel_m(\mathbf{X}_t) = \eta_m p(\mathbf{z}_t^m|\mathbf{X}) bel_m(\mathbf{X}_{t-1}) \tag{2.18}$$

$$bel_i(\mathbf{X}_t) = \eta_i p(\mathbf{z}_{i,t}|\mathbf{X}) bel_i(\mathbf{X}_{t-1}) \tag{2.19}$$

so

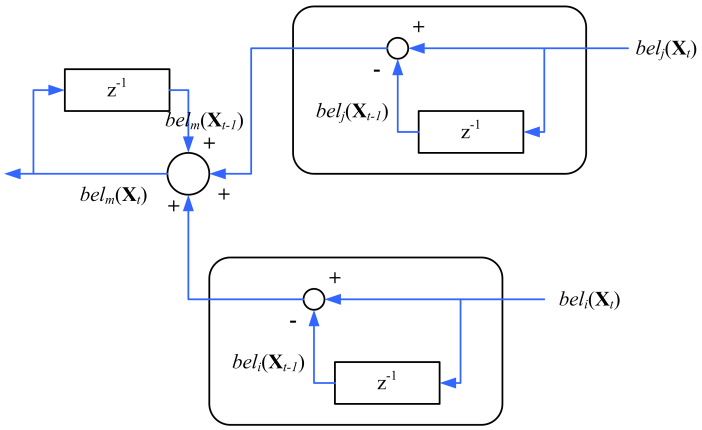

Figure 2.7: An scheme of the fusion procedure in the central node, in logarithmic form. The delays should be considered in a broad sense, and they represent the difference with the previous time when information was received.

$$bel_m(\mathbf{X}_t) = \eta bel_m(\mathbf{X}_{t-1}) \prod_{i=1}^{M} \frac{bel_i(\mathbf{X}_t)}{bel_i(\mathbf{X}_{t-1})} \tag{2.20}$$

Expressing the beliefs in logarithmic form:

$$\log bel_m(\mathbf{X}_t) = \log(\eta) + \log bel_m(\mathbf{X}_{t-1}) + \sum_{i=1}^{M} \left[\log bel_i(\mathbf{X}_t) - \log bel_i(\mathbf{X}_{t-1}) \right] \tag{2.21}$$

Therefore, what the central node has to do is to sum (in logarithmic form) into a running total the increase in *evidence* given by every robot of the fleet. Figure 2.7 shows a schematic representation of the fusion procedure. Even though no particular form or representation for the belief states is assumed yet, an interesting characteristic of this case is that, as the state is static, the message size for communicating the beliefs is fixed (Rosencrantz et al., 2003) (although depending on the belief representation and the dimension of the state, this number could be very high).

Finally, one of the most important issues is that, in this case, the beliefs can be received *asynchronously* and with *arbitrary latency* (i.e., in the previous equations $t-1$ can be substituted by the previous instant in which the central node received information from each robot). Hence, in the case of an static state, each robot can accumulate evidence and transmit it at convenience, without additional memory costs.

2.4.2 Dynamic environments

In the case of dynamic environments, similar fusion rules to (2.17) and (2.20) could be used to fuse the local filtering estimations $bel_i(\mathbf{X}_t)$. However, it will be seen that only under certain circumstances the fused belief is the same as the one that could be computed in a central node.

Consider the case that, even though the state is dynamic, the actions of the robots do not affect this state, and that the motion model $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ employed by the different robots is the same.

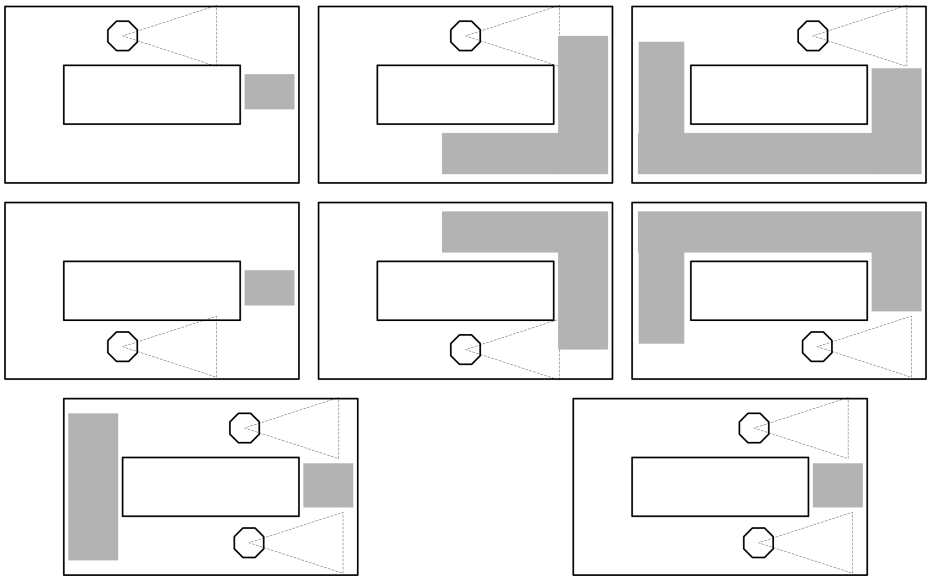

Figure 2.8: Two robots determining the position of an intruder. The two top rows represent the local belief state at times 0, 1 and 2 for the two robots. Third row, left: Combined belief state if robots transmit their beliefs at time 2. Right: global belief state at time 2 for the centralized case (adapted from (Rosencrantz et al., 2003))

Comparing equations (2.13) and (2.11), the relation between the complete belief and the local ones is given by:

$$bel_m(\mathbf{X}_t) = \eta \prod_{i=1}^{M} \frac{bel_i(\mathbf{X}_t)}{\prod_{i=1}^{M} \int p(\mathbf{X}_t|\mathbf{X}_{t-1}) bel_i(\mathbf{X}_{t-1}) d\mathbf{X}_{t-1}} \int p(\mathbf{X}_t|\mathbf{X}_{t-1}) bel_m(\mathbf{X}_{t-1}) d\mathbf{X}_{t-1} \qquad (2.22)$$

Equation (2.22) produces the same output as a centralized version (2.11) only if each robot sends its belief any time they update it with new data. Otherwise, information will be missed (Rosencrantz et al., 2003; Bourgault and Durrant-Whyte, 2004). Figure 2.8 shows an example. The first two rows show the evolution at times 0, 1 and 2 of the local belief state for the position of an intruder of two robots. The prior belief state at time 0 for the central node is the same. The third row, left, shows the obtained belief state $bel_m(\mathbf{X}_t)$ if the local belief states at time 2 are combined in the central node using (2.22). Clearly, this is different than the belief state that would be computed in a centralized system that receives all data at any time, represented by the the right scheme of the third row. Figure 2.9 summarizes the fusion equation in logarithmic form.

The problem is due to the fact that there are some informations not taken into account when performing the prediction steps in the fusing node. When the state is dynamic, the predicted belief state at any given time depends on all the past observations, so the predicted belief for a node with access to all information is not the same as the predicted belief for each individual robot. As noted in (Bourgault and Durrant-Whyte, 2004), these errors will increase mainly with the number

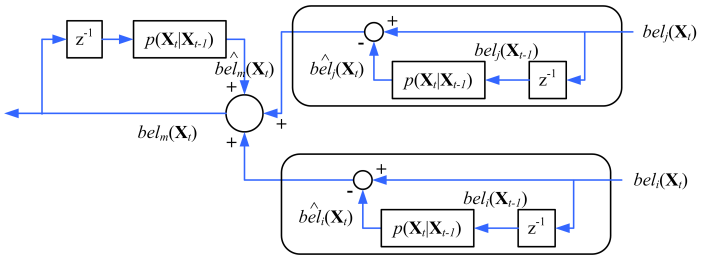

Figure 2.9: An scheme of the fusion procedure in the central node, in logarithmic form. In the dynamic case, delays in the transmission or missing information will lead to errors with respect to the optimal centralized estimation.

of predictions steps carried out between communication steps and with the change induced in the beliefs by the prediction (for instance, for fast changing states the effect will be bigger) and updating stages.

However, although eq. (2.22) is not optimal in the sense that the resultant belief state is not the same as in the centralized version, it can be seen that the resultant belief state is conservative, in the sense that the resultant belief state is consistent, as no redundant information is introduced.

In the dynamic case, there is a different option for the fusion of beliefs. (Rosencrantz et al., 2003) showed that, in the dynamic case, the role that was played before by belief state at time t, $bel(\mathbf{X}_t)$, is played now by the belief state over the full state trajectory up to time t, $bel(\mathbf{X}^t)$.

The belief state over the full trajectory for robot i is:

$$bel_i(\mathbf{X}^t) = p(\mathbf{X}^t|\mathbf{z}_i^t) = \eta^{'} p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} p(\mathbf{z}_{j,\tau}|\mathbf{X}_\tau) p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1}) \tag{2.23}$$

Comparing this expression to eq. (2.12) then:

$$bel_m(\mathbf{X}^t) = \eta p(\mathbf{X}_0^t) \prod_{i=1}^{M} \frac{bel_i(\mathbf{X}^t)}{p(\mathbf{X}_0^t)} \tag{2.24}$$

where $p(\mathbf{X}_0^t) = p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1})$.

Also, the same expression can be in recursive form:

$$bel_m(\mathbf{X}^t) = \eta p(\mathbf{X}_t|\mathbf{X}_{t-1}) bel_m(\mathbf{X}^{t-1}) \prod_{i=1}^{M} \frac{bel_i(\mathbf{X}^t)}{p(\mathbf{X}_t|\mathbf{X}_{t-1}) bel_i(\mathbf{X}^{t-1})} \tag{2.25}$$

Equation (2.25) is quite similar to the fusion equation (2.20), but over the (growing) belief on the state trajectory. Therefore, if each robot sends its belief state over the state trajectory, the beliefs can be combined to obtain a global one that would be equal to the one obtained in a centralized version. In the example of Fig. 2.8, if the robots share their beliefs about the full state trajectory, then they could be fused to obtain exactly the same estimation as the centralized case.

Moreover, as in the static case, the belief states can be received asynchronously. Each robot can accumulate evidence, and send it whenever it is possible to communicate with the central node.

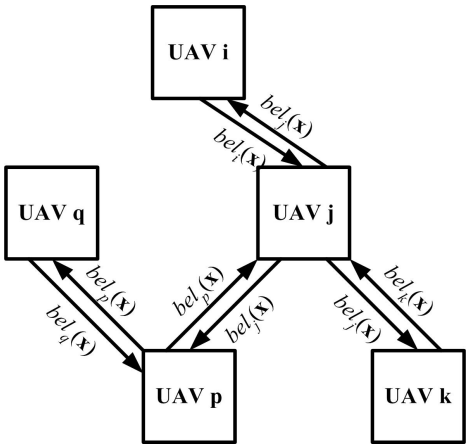

Figure 2.10: A group of UAVs forming a tree-shaped network. This topology allows removing the common information employing only local interactions.

However, the problem in this case is that the state grows over time, and therefore the size of the message needed to communicate the corresponding beliefs. In general, this will make this scheme of communication unaffordable. Nevertheless, for the normal operation of the robots, only the state trajectory over a time interval is needed, so these belief trajectories can be bounded. In any case, depending on the representation of the belief states and the dimension of the state itself, the size of information needed to be transmitted can be prohibitively high. However, it should be pointed out that, if the belief state over the trajectory is not communicated, in the dynamic case only approximations of the (ideal) centralized global belief state can be estimated.

2.5 Decentralized belief computation

The aimed scheme is that in which each robot locally estimates its belief by using a Bayes filter, and shares its local estimation with its companions. This scheme is called decentralized, and its characterized by the following restrictions: there is no central node that fuses the beliefs obtained by the robots; no robot has knowledge of the global network topology, and there is no global communication facility, in the sense that it cannot be ensured that the information transmitted by one robot will reach all the robots of the fleet. At any time instant, each robot will be able to communicate and share its belief directly only with a subset of the fleet (that could be the empty set).

If a robot was only a sink of information, then it could apply the fusion filters (2.20) and (2.25) described in the semi-decentralized case (in this case, the central node is the receiving UAV). However, the idea is that the robots collaborate, not only receiving information, but also communicating information with their neighbors. In this case, the previous filters should be modified in order to remove potential common information before incorporating the received belief[2]. This common

[2]The previous equations are already eliminating some common information due to the common prior $p(\mathbf{X}_0)$ and

information is due to the information previously sent to other robots.

There is a situation in which the common information can be, at least in theory, removed employing only local interactions. If the belief network topology [3] is a tree, as shown in Fig. 2.10, that is, if there is a unique path between any pair of providers and receivers, then the fusion equations (2.17) and (2.24) for the central node can be applied at every robot to fuse the beliefs received from its neighbors, with some modifications.

The aspect that must be considered is to avoid considering multiple times the same information through the direct link between two robots with direct communication (like UAV i and UAV j in Fig. 2.10). However, this can be solved locally, without further information about the rest of the network. Each robot only needs to know what information has previously been transmitted to its neighbors in order to remove it when integrating data received from them. If the common information that UAVs i and j share is represented by $bel_{ij}(\mathbf{X}_t)$, for the case of a static state, and expressing the beliefs in logarithmic form, the local fusion equation that UAV i uses to update its belief when it receives information from j is given by:

$$\log bel_i(\mathbf{X}_t) \leftarrow \log(\eta) + \log bel_i(\mathbf{X}_t) + \underbrace{\log bel_j(\mathbf{X}_t) - \log bel_{ij}(\mathbf{X}_t)}_{\Delta bel_{ij}(\mathbf{X}_t)} = \log(\eta) + \log bel_i(\mathbf{X}_t) + \Delta bel_{ij}(\mathbf{X}_t) \tag{2.26}$$

In the case of a tree topology, this common information can be estimated quite simply. As seen above, when UAV i receives information from UAV j, the information incorporated locally should not be considered the next time that information is received from the same UAV, and thus:

$$bel_{ij}(\mathbf{X}_t) = bel_j(\mathbf{X}_t) \tag{2.27}$$

In the same way, each time that UAV i sends its belief to UAV j, the evidence sent will be incorporated by UAV j, and therefore is part of the common information:

$$bel_{ij}(\mathbf{X}_t) = bel_i(\mathbf{X}_t) \tag{2.28}$$

The operations above performed constitute the so-called channel filter presented in (Bourgault and Durrant-Whyte, 2004) (although in a different formulation). The fusion scheme for the static case is represented in Fig. 2.11. In the case of a dynamic state, it is required to propagate the common information $bel_{ij}(\mathbf{X}_{t-1})$ to the current time instant, employing a prediction step, before removing this common information. It should be noticed that the previous equations are qualitative, in the sense that depending on the form of the belief states, the previous differences and the logarithms might not be defined.

However, the same scheme cannot be applied to general topologies without further considerations. Consider the scheme of connections of Fig. 2.12. In this case, there are multiple paths from the same UAV to others. The problem in this case is that the *same information* could be accounted

previous received information.

[3] The term network should be interpreted at the data level. That is, the underlying physical or transportation layers might have a different topology.

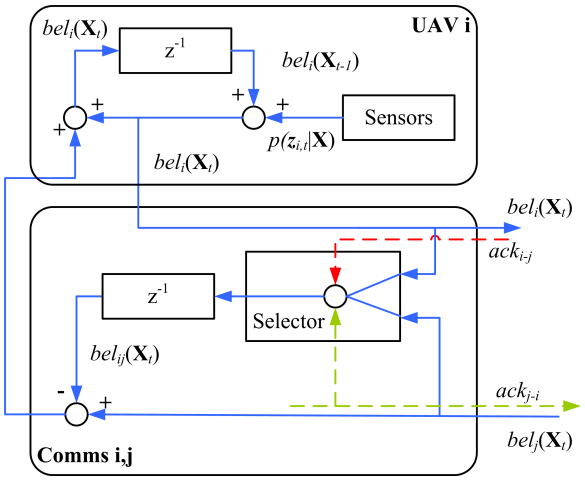

Figure 2.11: Scheme for UAV i in logarithmic form. The added blocks allow removing the common information. Each time i sends UAV j, the evidence transmitted will be subtracted from the information received from UAV j the next time. The same procedure is applied to the received information from j.

several times. Thus, the belief combination rule has to consider the common information (Grime and Durrant-Whyte, 1994; Julier and Uhlmann, 1997b; Sukkarieh et al., 2003). Otherwise, this could lead to overconfident estimates.

Without additional assumptions on the form of the belief distributions, their representations and others, there is no general solution to the problem of considering the common information in networks of arbitrary topology if only local information is considered (Utete and Durrant-Whyte, 1994). One solution would be to force a tree topology in the network. However, in a fleet of robots the topology of the connections is dynamic and the tree structure of the network would have to be ensured along time. Also, if a tree topology is forced, an interesting property is lost. One robot can act as a *data mule* between different subsets of robots of the fleet (see Fig. 2.13). When connected to one subset, the robot updates its belief from the information received from the subset. Then, when connected with the second subset, the robots of this second subset will incorporate information from the other subset indirectly through the belief state of the moving robot.

One important characteristic of the decentralized scheme is that, if the previous fusion rules are applied, in the ideal case all robots will finally converge to the same global belief about the state. Two main departures from the ideal case are to be considered. In the case of dynamic states, some kind of approximation is to be made in order to limit the bandwidth required to communicate the belief states over the trajectory. On the other hand, it is needed to consider potential overconfident estimations due to common information in the presence of loops in the information channels.

The previous sections have presented the general problem of information fusion for cooperative perception from a Bayesian point of view. The development of working algorithms, even in the case of centralized fusion, requires defining a particular representation of the belief states, which itself depends on the problem. The next chapters of the thesis present different approaches employing

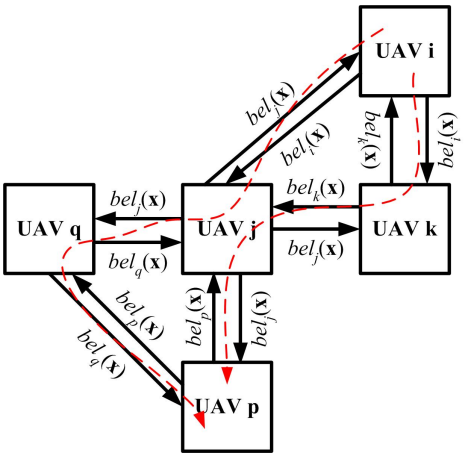

Figure 2.12: Multiple paths can lead to overestimation, when incorporating several times the same information.

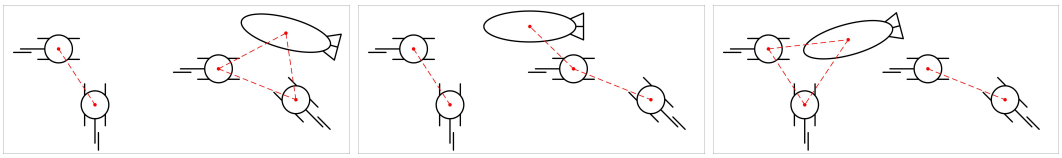

Figure 2.13: In the general case, the topology of the network will change with time. For instance, a robot can act as a data mule carrying data among different subsets of robots.

different assumptions for the form of the belief distributions and the transition and measurement models.

2.6 Developing cooperative perception actions

The previous sections have shown the main issues in decentralized information fusion. The final important aspect of cooperative perception is the ability to develop actions to improve the knowledge about the environment. The objective is to determine which actions $\mathbf{u}^m$ the robots should carry out in order to increase some measurement about the quality of the current knowledge.

Determining robots actions considering uncertain worlds is usually tackled employing decision theoretic techniques. The objective is to select actions considering the current knowledge of the world (represented by the belief state) and the possible outcomes from actions carried out by the robot. The computation of rational actions is considered including into the DBNs presented before a payoff or reward function on the state $R(\mathbf{x}), \mathbb{R}^n \longrightarrow \mathbb{R}$. This payoff value can consider costs and revenues for being in a particular state (see Fig. 2.14).

The objective is to find an optimal *policy*, that is, a function $\pi(\mathbf{x}) \rightarrow \mathbf{u}$, that determines what action $\mathbf{u}$ should be carried out if the state is $\mathbf{x}$. The policy should be optimal in the sense that it should maximize the expected cumulative payoff (also called expected utility):

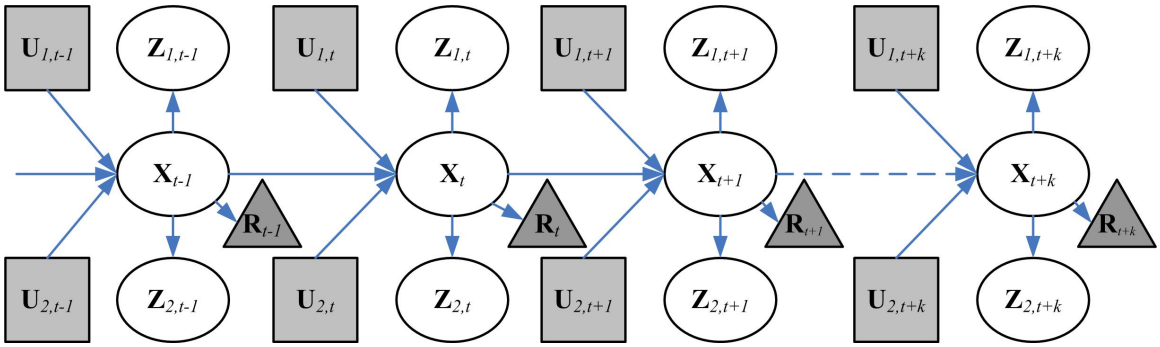

Figure 2.14: Planning and control under uncertainty is usually considered introducing a reward function on the state $\mathbf{R}_t(\mathbf{X}_t)$, which takes into account the benefits and costs of being in a particular state. Decision theoretic robots try to maximize the expected accumulative rewards $\mathbf{R}_t$.

$$\pi^* = \arg\max_{\pi} E\big[\sum_{t=0}^{T} \gamma^t R(\mathbf{X}_t)|\pi\big] \tag{2.29}$$

that is, the expected sum of the rewards from time 0 to time T (T is called the planning horizon and $\gamma \in [0, 1]$ is the discount factor).

If the environment is fully observable (that is, if we are certain about the state any time we get new information), but the effect of the actions carried out is non-deterministic, the problem is called a *Markov Decision Process* (MDP) (Russell and Norvig, 2003). In the more general case in which the environment is not observable, the problem is called *Partially Observable Markov Decision Processes* (POMDPs)(Smallwood and Sondik, 1973).

Although usually employed for planning and control, choosing a convenient payoff function, POMDPs can be also used for developing perception actions; in this case, the payoff function usually considers higher values for narrower belief states, meaning that more informative belief states are preferred. However, the POMDP framework is usually unfeasible in this case, as the number of unknowns is high. The problem is even more important in the case of multiple robots, in which the problem to be solved is to determine an optimal policy that computes the actions of all the robots of the fleet $\mathbf{u}^m$ given the global belief state $bel_m(\mathbf{X})$.

Another option for developing cooperative perception actions is to define a measurement of the information gain obtained when executing a certain information gathering task. Several different measurements of the information gain can be used. For instance, for unimodal distributions, the covariance could give a measure of the amount of information gained when executing an action (the bigger the covariance, the more uncertain about the actual value of the state). Another and more general measure about the information of a probability distribution is the entropy. The entropy H of a probability distribution $p(x)$ is defined as the expected value of the information $-\log[p(x)]$. That is, the entropy of the belief state is given by:

$$H(t) = E_{\mathbf{X}_t}[-\log bel(\mathbf{X}_t)] = -\int bel(\mathbf{X}_t) \log bel(\mathbf{X}_t) d\mathbf{X}_t \tag{2.30}$$

The information gain is defined as the variation in the entropy after carrying an action $\mathbf{u}_{t+1}$. When executing this action, a new belief state $bel(\mathbf{X}_{t+1})$ will be obtained from the measurement $\mathbf{z}_{t+1}$ received, and the new entropy will be:

$$H(\mathbf{u}_{t+1}, \mathbf{z}_{t+1}) = -\int bel(\mathbf{X}_{t+1}) \log bel(\mathbf{X}_{t+1}) d\mathbf{X}_{t+1} \tag{2.31}$$

However, only the action $\mathbf{u}_{t+1}$ can be selected. Therefore, the expected entropy considering the potential measurements can be computed in advance:

$$\begin{aligned} E_{\mathbf{Z}_{t+1}}[H(\mathbf{u}_{t+1}, \mathbf{Z}_{t+1})] = \int H(\mathbf{u}_{t+1}, \mathbf{Z}_{t+1}) p(\mathbf{Z}_{t+1}) d\mathbf{Z}_{t+1} = \\ = \int H(\mathbf{u}_{t+1}, \mathbf{Z}_{t+1}) \int p(\mathbf{Z}_{t+1}|\mathbf{X}_{t+1}) \int p(\mathbf{X}_{t+1}|\mathbf{X}_t, \mathbf{u}_t) bel(\mathbf{X}_t) d\mathbf{X}_t d\mathbf{X}_{t+1} d\mathbf{Z}_{t+1} \end{aligned} \tag{2.32}$$

Therefore, the information gain associated to action $\mathbf{u}_{t+1}$ is defined as follows:

$$I(\mathbf{u}_{t+1}) = H(t) - E_{\mathbf{Z}_{t+1}}[H(\mathbf{u}_{t+1}, \mathbf{Z}_{t+1})] \tag{2.33}$$

This metric can be used to establish preferences among actions. A policy that at the same time maximizes the information gain and minimices action costs can be used for developing cooperative actions for perception purposes.

Both, the POMDP approach and the information gain-based approach can be used in a wide range of action selection problems, from optimal control to planning activities.

The decisional architecture for multi-robot coordination employed in the multi-robot fleet of the COMETS project (Gancet and Lacroix, 2004; Gancet et al., 2005) deals with the topics of multi-robot task planning and allocation, supervision and coordination. It can consider coordinated information gathering tasks, as cooperative detection and confirmation and cooperative monitoring. It does so by means of predefined roles and off-line learned models of performance by the different UAVs (like detection rates and precision in localization tasks). Also, the decisional layers can refine their plans taking into account perception-based refiners that reason about the current knowledge and the perception capabilities of the UAVs. Therefore, in this thesis this topic is not further considered in depth. More information about this decisional architecture will be shown in Chapter 8, and can be found in (Gancet et al., 2005).

2.7 Related work on multi-robot cooperative perception

The research on multi-robot systems has extended for nearly the last twenty years. Several surveys summarize the work on the late eighties and early nineties, as for instance (Dudek et al., 1993; Cao et al., 1997). More recent surveys are (Parker, 2000; Balch and Parker, 2002). The interest in the field has grown in the last ten years, as can be seen in several Special Issues of different Journals (see Table 2.2[4]) and particular conferences and symposiums on the topic.

[4]This table does not pretend to be exhaustive

Table 2.2: Some special issues on Multi-Robot Systems

Journal	**Issue**	**Date**
Autonomous Robots	Vol. 8. Iss. 3	June 2000
Int. J. Robotics and Automation	Vol. 16, Iss. 4	2001
IEEE Trans. on Robotics and Automation	Vol. 18, Iss. 5	Oct. 2002
IEEE Trans. on System, Man and Cybernetics, Part A	Vol. 33, Iss. 5	Sept. 2003
Autonomous Robots	Vol. 17, Iss. 1	Jul 2004
Robotics and Autonomous Systems	Vol. 50, Iss. 2-3	Feb. 2005
Proceedings IEEE	Vol. 94, Iss. 7	Jul. 2006

2.7.1 Multi-robot architectures

An important part of the work on multi-robot systems has been focused of the development of architectures for robot coordination. For instance, ACTRESS (Asama et al., 1989), CEBOTS (Ueyama and Fukuda, 1993), ALLIANCE and L-ALLIANCE (Parker, 1998), and others. In general, they focus on task planning (Alami et al., 1998b), task allocation, coordination, planning and control (Feddema et al., 2002), conflict resolution, etc.

Although a group of robots can coordinate with no or little communication (Mataric, 1992; Balch and Arkin, 1998), in general, the previous architectures require communication of actions, plans, tasks, which can be interpreted as a kind of information sharing. However, apart from that, the knowledge sharing problem and its implications on robot coordination is not explicitly considered in most of those approaches (some exceptions are the ACTRESS architecture and the CEBOTs).

More recent architectures consider information fusion. Fierro et al. (2002) present an architecture and a high-level language, CHARON, for multi-robot coordination. The software architecture allows for hierarchical and distributed implementation of planning and perception activities. The system has been demonstrated in mapping activities in indoor environments, but the fusion of data is centralized, and the techniques employed for fusion are not described.

A decisional architecture similar to that developed in the COMETS project is the work of Simmons et al. (2002), which extends their three-layer robot architecture to the multi-robot case. The layers of the different robots can interact, so that the robots can exchange information at different levels (tasks, plans, etc). Information representation and fusion is not explicitly addressed; however, at the lower (the behavioral) level, robots can form distributed control loops in which the inputs to one robot skill can come from a different robot, which can also be seen as a type of information sharing. This kind of tight cooperation is not addressed in COMETS.

In the software agents community there have been efforts for standardizing knowledge-sharing among communicating agents. Several languages and communication protocols have been developed, like KIF, KQML[5] (Finin et al., 1997), although in general, the underlying knowledge representation is based on first order logic and does not consider the representation of uncertainty.

[5]KIF stands for Knowledge Interchange Format, and KQML for Knowledge Query and Manipulation Language

2.7.2 Multi-robot teams in cooperative perception activities

Of course, there have been applications of robot teams in cooperative perception activities. In this section some of them are presented, classified by the particular application. Then, they will be classified with respect to the techniques employed.

Mapping

One of the most common applications of multiple robots is mapping and exploration, as clearly they offer advantages over the simple robot case, for instance in the time needed to cover an unknown area (with time savings that can be superlinear in the number of robots).

One of the first applications of multiple robots for mapping is presented by Singh and Fujimura (1993). In (Grabowski et al., 2000), an heterogeneous team of robots, the Millibots, is employed in mapping activities. The paper also shows how specialization (robots specialized for a particular tasks) and cooperation can be used to overcome certain limitations. In (Thrun, 2001), a probabilistic method to combine local maps built by several robots is presented. Other examples of robot teams for mapping include (Simmons et al., 2000; Shuo and Min, 2000; Madhavan et al., 2004; Burgard et al., 2005).

Most of the previous systems consider only 2D maps. Less research can be found on cooperative mapping in 3D. See, for instance, (Rocha et al., 2005).

Most systems employ metric representations. Several approaches for cooperative mapping based on topological maps has been also presented, as in (Dudek et al., 1998; Dedeoglu and Sukhatme, 2000; Huang and Beevers, 2005).

Localization and SLAM

For cooperative activities in a robot team, it is very important that the robots are localized in a common frame. There are approaches to cooperative localization for a teams of robots. For instance, in (Kurazume and Nagata, 1994), where some of the robots, which remain static, act as landmarks for the others. Afterwards, the roles of the two subsets of robots are exchanged. The paper presents a study in simulation comparing the errors committed with those obtained using only local odometry. A similar approach for two robots is presented in (Rekleitis et al., 2003), where observing the companion is used to reduce the uncertainty on the position of the robots. Other results on cooperative localization are for example (Roumeliotis and Bekey, 2000; Fox et al., 2001; Spletzer and Taylor, 2003a; Madhavan et al., 2004).

More recently, approaches have appeared including cooperative multi-robot Concurrent Mapping and Localization[6] (CML). And thus, in (Fenwick et al., 2002) the benefits of cooperative CML are analyzed from a theoretical point of view. Other example is (Thrun et al., 2004).

Howard et al. (2006) have described results within the DARPA Software for Distributed Robotics initiative. An heterogeneous team of near 80 robots in indoor experiments is considered. Mapping activities, SLAM and cooperative detection of intruders are applications considered.

[6] Also Simultaneous Localization And Mapping (SLAM)

Tracking, surveillance

Other applications include multi-robot surveillance. For instance, the objective of the CyberScout project (Saptharishi et al., 2002) is the development of a network of robots for reconnaissance and surveillance operations. Stoeter et al. (2002) present a software architecture based on CORBA for multi-robot surveillance, although little emphasis is put on information sharing. Other example is the work of Lum et al. (2006) .

Multi-robot target tracking is considered in several multi-robot systems, like the BEAR project (Vidal et al., 2002b), which addresses pursuit-evasion games with aerial and ground robots. The problem of cooperative object tracking is also considered in (Rosencrantz et al., 2003; Stroupe et al., 2004).

In the context of Robocup (Kitano et al., 1998), several cooperative perception results have also been presented. Robocup is a testbed for cooperative algorithms for action and perception purposes. The teams clearly can benefit from information sharing. Typical applications is cooperative estimation of the ball, team mates and opponents position in robotic soccer, as for instance (Weigel et al., 2002). Other examples can be found in (Stroupe et al., 2001; Schmitt et al., 2002; Pagello et al., 2006).

Most previous approaches are applied in structured environments. Regarding unstructured environments, less work can be identified. Also, cooperative systems employing multiple UAVs are more scarce (they will be presented below). Many of the applications employ ad-hoc algorithms for combining data from several sources. The different approaches differ in the way they represent the information, how the data is communicated and fused and the system architecture (centralized or distributed). In the next sections, a classification of some of these approaches will be attempted.

2.7.3 Information representation and fusion

Bayesian-based approaches

Regarding information representation and fusion, most of the systems described above employ probabilistic representations for dealing with uncertainty, and Bayesian approaches for data fusion.

The most extended tools are Gaussian distributions and Kalman Filters, for instance for target tracking (Stroupe et al., 2001), localization and SLAM (Roumeliotis and Bekey, 2000; Fenwick et al., 2002; Thrun et al., 2004). In some applications, Multiple Hypothesis Tracking (MHT) (Reid, 1979) is used to overcome some of the problems of the simple Kalman filter, like data association in clutter (Schmitt et al., 2002).

Kalman filters assumptions are usually violated in robotics, where multi-modal distributions and non-linear measurement and transition models are common. Therefore, non-parametric representations of the belief state have also been considered for multi-robot estimation.

Grid-based representations (Elfes, 1989) are also used for multi-robot perception, as in (Singh and Fujimura, 1993; Grabowski et al., 2000; Thrun, 2001; Rocha et al., 2005) in cooperative mapping applications. In (Weigel et al., 2002) a mixture of Kalman filtering and grids is used for ball position

determination in Robocup. Also grids are used as environment representation in multi-robot pursuit-evasion games in (Vidal et al., 2002b).

In the last years, the use of *particles* as non-parametric representations of the belief and the use of *particle filters* for state estimation (Doucet et al., 2001) has become a common tool in robotic applications. The use of particles in multi-robot teams has been considered more recently. In (Rosencrantz et al., 2003), the use of particle filters for decentralized sensor fusion in dynamic environments is described. Simulation results with fleets of tenths of robots are presented, along with preliminary results with several robots, in indoor environments. This work, however, does not considers the problem of removal of common information explicitly. Another example on the use of particle filters for multi-robot perception is the work of Ong et al. (2005b).

Non-bayesian approaches

A common alternative approach for uncertain reasoning is the Dempster-Shafer theory of evidence (Dempster, 1968; Shafer, 1976), that tries to deal with the issue of ignorance, as opposed to uncertainty.

In Dempster-Shafer theory, given the possible values of the state (the frame of discernment Ω, for instance the two potential values of a cell in an occupancy grid, empty -*E*- or occupied -*O*-), the degree of belief is represented by a real number in $[0, 1]$ associated to each potential hypothesis. The Basic Probability Assignment is a function $m : \Psi \rightarrow \quad [0, 1]$, where Ψ is the set of all possible subsets of Ω, $\Psi = 2^{\Omega}$ (in the example, $\Psi = \{\emptyset, E, O, EO\}$, where EO implies unknown and $\emptyset$ is the empty set). The belief m is defined so that $\sum_{A \subseteq \Psi} m(A) = 1$ and $m(\emptyset) = 0$.

Given two belief functions m_1 and m_2 over the same frame of discernment but provided by independent sources, the Dempster rule allows obtaining the global belief:

$$m_1 \oplus m_2(A) = \frac{\sum_{B \cap C = A} m_1(B) m_2(C)}{1 - \sum_{B \cap C = \emptyset} m_1(B) m_2(C)} \tag{2.34}$$

$$m_1 \oplus m_2(\emptyset) = 0 \tag{2.35}$$

Dempster-Shafer theory has been also employed for multi-robot perception. For instance, Cai et al. (1997) present an evidential reasoning approach to multi-robot information sharing in the CEBOT system, applied to multi-robot mapping. Results are shown in simulation. The authors discuss the advantages and drawbacks of evidential reasoning with respect to Bayesian approaches. Another example is (Shuo and Min, 2000), where the authors extend the algorithm for map-building of Pagac et al. (1998), based on Dempster-Shafer belief functions, to the multi-robot case. Some results in simulation are presented. Yu et al. (2004) present an analysis in simulation of evidential approaches for airborne recognition in military applications. Also, in (Yanli et al., 2005), the authors present a multi-robot map-building approach based on evidential reasoning.

However, issues like decentralized schemes, rumor propagation, etc. are not covered in general by these approaches.

Other possibility for uncertain reasoning is possibility theory (Zadeh, 1978), built over the arithmetic of fuzzy sets. Some authors have presented results on information fusion using possibility

theory, as for instance in (Cánovas et al., 2004), where the authors employ it for cooperative localization and ball position estimation in Robocup, but again nothing is commented about the extension of these ideas to decentralized systems.

Another interesting approach has been presented by Spletzer and Taylor (2003a) in the framework of multi-robot localization. In this work, uncertainty in some quantities, as range and bearing measurements, is represented as bounding intervals. These intervals induce a polytope in the configuration space that represent the set of configurations compatible with the sensor measurements. These polytopes can be combined in order to obtain the resultant belief about a certain quantity when several sources of information are available.

Axiomatic approaches for combining probability distributions

In the field of Management Science, the problem of combining expert opinions has been researched for the last fifty years. The main topic is how a decision maker should fuse a set of expert opinions in order to take the right decisions (see (Genest and Zidek, 1986) for an early review). These expert opinions are expressed as probability distributions about certain quantities. The framework employed is also called Consensus Theory.

A common approach for combining the expert opinions is to use what is called an opinion pool. The objective is that a decision maker obtains a unique pooled opinion about $\mathbf{X}$ from a set of individual opinions from experts, represented by their beliefs $bel_i(\mathbf{X})$. Each expert is locally Bayesian. That is, each expert applies the Bayes filter for estimating its own belief. The global central belief is obtained by using a combination rule T

$$bel_m(\mathbf{X}) = T[bel_1(\mathbf{X}), \ldots, bel_i(\mathbf{X}), \ldots, bel_M(\mathbf{X})] \tag{2.36}$$

The combination function T is usually deduced from a set of axioms, like exchangeability of opinions (Morris, 1983). Other approaches introduce a loss function $L[bel_m(\mathbf{X}), \mathbf{x}^*]$, defined as the loss induced by adopting $bel_m(\mathbf{X})$ when the actual value of the state is $\mathbf{x}^*$. Then, fusion rules that minimize the expected loss $E\{L[bel_m(\mathbf{X}), \mathbf{x}^*]\}$ for all potential values of $\mathbf{X}$ are sought (DeGroot and Mortera, 1991).

Usually, a family of combination rules are considered. For instance, the so called Linear Opinion Pools (LOPs), in which the combination rule is a weighted average:

$$bel_m(\mathbf{X}) = \sum_i w_i bel_i(\mathbf{X}) \tag{2.37}$$

In Logarithmic Opinion Pools (LGPs), the combination rule is:

$$bel_m(\mathbf{X}) = \eta \prod_i bel_i^{w_i}(\mathbf{X}) \tag{2.38}$$

It can be seen that for the case of conditional independent observations given the state, LGPs give the same fusion rules as presented in Section 2.4. This case is called the *independent opinion pool*.

These techniques approach the problem of rumor propagation in an axiomatic way, as the final fusion or consensus rule T can take into account correlations between the experts. Also, they can tackle with disagreement, which is not considered in the previous framework. Nevertheless, decentralization of this kind of algorithms is an open problem. These approaches were largely ignored within the multi-robot research until very recently (Blanco et al., 2007; Pahliani and Lima, 2007).

2.7.4 Centralized vs decentralized approaches

Many of the previous systems described above employ a centralized architecture for information fusion. As an example, the CS Freiburg team in Robocup (Weigel et al., 2002) uses a centralized node for data fusion. Also, hierarchical architectures are considered. In the Millibots (Grabowski et al., 2000) there are special leaders that receive all the information from the other robots (the Millibots do not have enough processing power for mapping) and performs perception.

The work of Howard et al. (2006) considers a hierarchical scheme for information fusion, applied for localization and mapping. Each robot of the fleet performs locally an SLAM algorithm for localization and mapping. Then, each robot sends its pose and sensor readings to a central node, and there all data are fused to obtain a global and consistent map. Interestingly, each robot can recognize, identify and determine the relative pose of any other within its field of view. This information is used as constraints for building a common map.

In this thesis, a decentralized approach is followed. The main aspects of decentralized information fusion have been considered in the sensor network community since the last decade (Bar-Shalom and Li, 1995). A review on distributed fusion architectures and algorithms can be found in (Liggins II et al., 1997). These papers state the main problems in decentralized estimation, considering the general Bayesian solution for the case of static states and the problem of removing common information in general distributed networks. In the initial approaches, in most cases only Gaussian linear systems are considered, and non-linear systems are tackled through linearization. Kalman filters (and Extended Kalman Filters) are the main tools.

The group of Durrant-Whyte has been one of the leading groups on decentralized state estimation in automation. The use of decentralized sensor fusion for a process plant monitoring is presented in (Grime and Durrant-Whyte, 1994). In this work, the Information Filter (dual of the Kalman Filter) is used as the main tool for data fusion. The problem of common information in networks of general topology is identified, and it is stated that no solution for this is available with only local network knowledge. In (Utete and Durrant-Whyte, 1994), network management policies are considered to ensure the correct functioning of distributed fusion.

The extension of the previous ideas to robotics has been considered also in this group. In (Mutambara, 1998; Mutambara and Durrant-Whyte, 2000), they present the application of decentralized state estimation and decentralized control of a single wheeled robot. However, the application of the previous ideas for multi-robot cooperative perception has appeared more recently.

In (Roumeliotis and Bekey, 2000), a distributed Kalman filter is employed for fusing relative position measurements between robots for improving the estimation of the position of all the robots

of the fleet. Very interestingly, the paper shows how not considering common information can lead to overconfident estimates.

Decentralized algorithms have been proposed for the problem of multi-robot Simultaneous Localization And Map building (multi-robot SLAM). The SLAM problem poses the additional constraint that the state is high-dimensional. Fenwick et al. (2002) use a decentralized Kalman filter for map fusion, although the problems of rumor propagation are not considered. In (Thrun et al., 2004), a probabilistic approach for SLAM using the Information Filter (see Chapter 3) is presented. The sparseness properties of the information form representation of the belief are exploited to reduce the bandwidth needed to communicate the belief among robots. However, the fleets considered are small, and small emphasis is made on problems like communication, rumor propagation, etc.

The techniques described in (Spletzer and Taylor, 2003a) for cooperative tracking are claimed to be decentralized, although the paper is centered on the configuration of the robots and not in the information sharing problem.

A decentralized architecture is also considered in (Rocha et al., 2005) for 3D mapping. The robots share position and visibility data for coordination in exploration. They also share part of the set of measurements collected for mapping (the most useful measurements, defined using information theory criteria).

Also, Howard et al. (2006) report results on distributed detection of intruders using acoustic sensors by a fleet of robots. The algorithm for integration of information is quite simple (but effective) and non-probabilistic in nature. In (Pagello et al., 2006), a distributed approach based on the Kalman filter is employed for ball tracking in Robocup. However, the issues of rumor propagation are not considered.

Rosencrantz et al. (2003) present a decentralized approach to multi-robot perception employing particle filters. Recently, a comparison of different representations and their implications for decentralized data fusion has been presented in (Ong et al., 2005a).

2.7.5 Cooperative perception actions

The development of coordinated information gathering actions has been considered in some multi-robot systems.

One of the applications more related to the development of cooperative perception actions is multi-robot exploration of unknown environments. In these problems, a group of robots should obtain a map. The main issue is to determine how to move the robots to cover the area, usually minimizing the time. Several approaches have been presented. For instance, Burgard et al. (2005) employ a decision-theoretic approach for robot coordination. As in (Yamauchi, 1998), the potential points where to move the robots are the so-called frontier cells, points in the border of unexplored areas that are likely to provide a high information gain. The utility of these frontier cells depends on the cost of travelling to them and the number of robots on the vicinity. A centralized approach is used to determine an optimal assignment of robots to cells. In (Simmons et al., 2000), a semi-decentralized scheme is used to coordinate a fleet of robots in an exploration task. Each robot sends bids to a central executive. Each bid consists of a set of frontier cells along with their associated costs. Again,

these costs depend on the distance traveled and the expected information gain. This information gain is computed as the expected number of sensor-covered cells in a grid environment, considering sensor ranges and current obstacles. Coordination is achieved by including a discounting factor in the expected information gain when several robots propose the same frontier cell, thus preventing various robots to explore the same region.

In (Stroupe et al., 2004), an algorithm called Most Value Estimation for Robot Teams (MVERT) for multi-robot action selection is presented. The technique tries to compute the utility of a robot over a full set of tasks. Cooperative perception is considered including exploration tasks, and the covariance matrix is used as a measure of the information (as Gaussian distributions are used as representations).

A closer approach to the decisional architecture employed in the fleet of robots considered in this thesis is that presented by Zlot et al. (2002). In this approach, the robots coordinate in an exploration task employing a market-like negotiation (Dias and Stentz, 2000). Goal points for unexplored areas are generated, and the different robots submit bids. These bids consider the cost for each robot to accomplish the task (in general, a distance-based cost) and also the revenue, again related with the expected *information gain*. This information gain is estimated as the expected number of new cells explored, considering a grid-based representation of the environment. The coordination is decentralized, and each robot computes its bids considering local information.

In (Parker, 2002), the problem of Cooperative Multi-Robot Observation of Multiple Targets or CMOMMT is defined as a maximization problem, in which a fleet of robots tries to maximize the number of targets being observed at a certain time.

While the previous approaches deal with task planning and allocation, most of the approaches for multi-robot exploration are more related to optimal control. They focus on the computation of optimal trajectories from the point of view of information gain. This is the case of the valuable work by Grocholsky et. al (2003). In this work, they present the computation of optimal trajectories from an information-theoretic formulation of utilities. They distinguish between coordination and active cooperation, the later involving explicit negotiations between robots in order to obtain a global optimum for the fleet, and that should also consider aspects from game theory. In a coordinated solution, all decisions are taken locally, but, as they show in the paper, the fact that information is shared among robots produces a sort of coordination among the robots. The analysis of the case of explicit cooperation is also covered in (Grocholsky, 2002). This is also discussed in (Bourgault et al., 2004; Wong et al., 2005) for the case of target search by a set of robots. It is important to notice that, in this case, the decision making process is decentralized. Also, Spletzer and Taylor (2003b) present a decentralized approach for optimal multi-robot motion for target tracking.

2.7.6 Middleware

Cooperative perception requires adequate communication layers for information sharing. In the last years, the development of middlewares for multi-robot communication has boosted the research in multi-robot teams. In (Utz et al., 2002), a CORBA-based middleware for cooperative robotics is presented. It has been proposed for its use in robotic soccer. Also, Chaimowicz et. al (2003)

present a programming framework (based on the .NET Framework) for multi-robot perception. This framework has been used for aerial-ground robot applications (Chaimowicz et al., 2004).

Player/Stage (Gerkey et al., 2003; Collett et al., 2005) has become a very popular tool and framework for multi-robot development. The Player server is used as a hardware abstraction layer that can be used with many real robots or simulated ones, and it includes communication facilities through TCP sockets that allow robot intercommunication. Moreover, it has been used in some projects related to cooperative perception (Grocholsky et al., 2003). However, the use of TCP could be a problem in wireless outdoor environments.

Another popular tool employed in the last years is YARP (Yet Another Robot Platform). YARP (Metta et al., 2006) offers facilities of communication between modules over TCP and UDP, and can be used for multi-robot developments.

In this thesis, the BlackBoard Communication System (BBCS) (Remuß and Musial, 2004; Remuß et al., 2004), developed by the TUB is used as communication layer. This will be described more in detail in Chapter 8.

2.8 Cooperative perception in aerial robotics

The cooperative perception techniques will be applied to fleets of Unmanned Aerial Vehicles (UAV). The related work on UAVs is presented in this section.

2.8.1 UAV research

As commented before, UAVs have been mainly used for military purposes and, in general, for classified purposes. During the last decades significant efforts have been devoted in the military field to increase the flight endurance and payload of UAVs. UAVs can be classified according to these parameters as:

- High Altitude Long Endurance (HALE) UAVs, as for example the Northrop Grumman Ryan's Global Hawks (65000 ft altitude, 35 hours flight and 1900 lbs. payload).
- Medium Altitude Long Endurance (MALE) UAVs, as for example the General Atomics' Predator (see Figure 2.15, left, with 27000 ft altitude, 30/40 hours flight and 450 lbs payload)
- Tactical UAVs such as the Pioneer with 15000 ft altitude, 5-6 hours flight and 25 Kg. payload

In the last years, man portable or hand launched UAVs, called "Organic UAVs" , such as Pointer (AeroVironment), Javelin (BAI) or Black Pack mini (Missions Technologies) have been presented. Furthermore, many different Vertical Take-Off and Landing UAVs including helicopters and several new designs such as the Guardian from Bombardier, and the Sikorksys Cypher or Dragon Warrior (see Figure 2.15, centre) have appeared. On the other hand, Micro Air Vehicles, with dimensions lower than 15 cm, have gained a lot of attention recently. These include the Black Widow manufactured by AeroVironment (see Fig. 2.15, right) or the MicroStar from BAE.

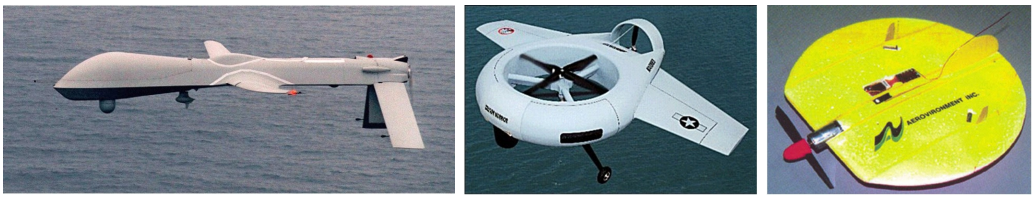

Figure 2.15: Left: The Predator from General Atomics. Centre: The Sikorskys Dragon Warrior (Cypher2). Right: Black widow (AeroVironment).

Figure 2.16: Left: UAV Platform of the WITAS Project. Right: Autonomous blimp Karma developed at LAAS.

In the civil area, in the last fifteen years a significant progress towards autonomous aerial vehicles with on-board intelligent capabilities has been experienced. The Universities and Research Centres in the United States of America (USA) have played an important role. For example, the University of Southern California (USC) has conducted, since 1991, an autonomous helicopter project developing several prototypes, such as the AVATAR (Autonomous Vehicle Aerial Tracking and Retrieval/ Reconnaissance), and a behaviour-based architecture for helicopter control (Fagg et al., 1993; Montgomery et al., 1995). The AVATAR helicopter won the AUVSI Aerial Robotics competition in 1994.

Georgia Tech. also has a significant tradition in aerial robotics, being very active in the International Aerial Robotics Competition (Christophersen et al., 2001), and developing autonomous helicopters and research in flight control, avionics and software systems.

The Carnegie Mellon University Robotics Institute also has more than ten years of expertise on autonomous visual-based helicopters with on-board vision systems (Amidi et al., 1999).

Other research efforts in autonomous aerial vehicles are carried out at the University of California at Berkeley, with the BEAR project (Vidal et al., 2002a), and at the University of Florida (Stancliff and Nechyba, 2000), which is developing an UAV platform and techniques for learning to fly from human pilots.

There is also significant autonomous aerial vehicle research in Australia, mainly at the Sydney University, where several projects involving UAVs have been developed in the last years (Sukkarieh et al., 2003; Kim and Sukkarieh, 2004).

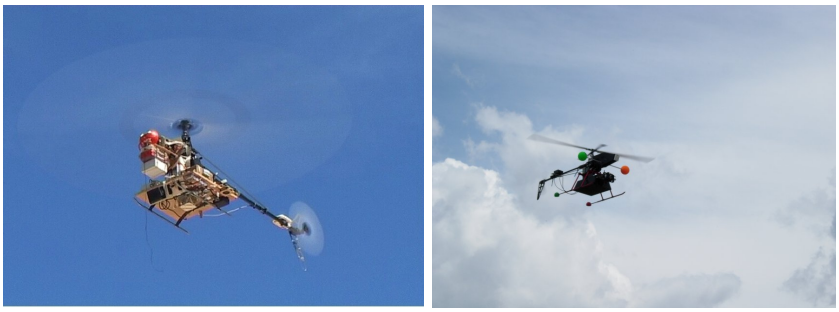

Figure 2.17: The autonomous helicopter Marvin in two stages of its evolution. Left: Marvin I. Right: Marvin II.

In Japan, in the early nineties MITI funded several robotic helicopter projects, including the autonomous helicopter at the Tokyo Institute of Technology, which demonstrated autonomous capabilities by using fuzzy logic control (Sugeno et al., 1993).

In Brazil, the project AURORA aims at the development of an unmanned airship capable of autonomous flight over user-defined locations for aerial inspection and image acquisition (Bueno et al., 2002).

Finally, it is well known that Israel has also a long tradition on the development of UAVs mainly for military applications.

The activities in aerial autonomous systems in Europe are scarce when comparing with the efforts in USA, although in the last years there has been an increase in the number of prototypes and projects.

In Sweden, the WITAS project, already finished, dealt with the development of an autonomous helicopter and several UAV technologies (Granlund et al., 2000; Merz, 2004). Their prototypes are based on the R-MAX platform of Yamaha (see Figure 2.16 left). Also, several Universities such as ETH Zurich (Eck et al., 2001), and Universidad Politécnica de Madrid (Del Cerro et al., 2002) are using the adaptation of conventional radio controlled helicopters with different autonomous capabilities.

Special mention should be given to Technical University of Berlin, who has developed a very successful autonomous helicopter demonstrating autonomous take-off, object location and retrieval and landing (Musial et al., 2000) (Remuß et al., 2002), and which won the AUVSI Aerial Robotics Competition in 2000 (see Fig. 2.17).

At the LAAS (Toulouse) an autonomous airship has been developed (see Figure 2.16 right). The research interests of this group include one of the few SLAM applications for an aerial vehicle (Hygounenc et al., 2004). An autonomous airship is also considered by the University of Stuttgart (Wimmer et al., 2002).

At the University of Seville, several prototypes have been built in the last years. A conventional remotelly controlled helicopter was converted for the COMETS project, by adding sensing, perception, communication and control functions (Ollero et al., 2003). Also, the HERO autonomous helicopter is under development (see Fig. 2.18).

Figure 2.18: The HERO helicopter developed at the University of Seville

2.8.2 Multi-UAV systems

Over the past few years, research on the coordination and cooperation of multiple UAVs and of multiple aerial and ground autonomous systems has been conducted all over the world.

Several efforts are related to the coordination of homogeneous teams of aeroplanes (McLain, 2000). The problems are related to the control of multiple UAVs (aeroplanes) in close-formation flight, as for example in the Air Force Research Laboratory (Schumacher and Singh, 2000); or in the Air Force Institute of Technology (Hall and Pachter, 1999). The Phoenix Project at Princeton University also considers the coordinated flight of a fleet of homogeneous UAVs (aeroplanes) and the design scenario is autonomous aerobatic manoeuvring. The problem of autonomous formation flight control is also considered in (Giulietti et al., 2000).

At MIT, the Aerospace Controls Laboratory[7] also conducts research on coordination of multiple UAVs (How et al., 2004). A Multi-vehicle Experimental Platform for Distributed Coordination and Control is being developed. This project will build a multi-vehicle testbed to demonstrate and evaluate the coordination and control approaches under development at MIT.

Multiple flying helicopters and groups of helicopters and ground vehicles are considered in the previously mentioned BEAR project. The research includes hierarchical multi-agent system architectures for coordinated team efforts, vision-based pose estimation of multiple UAVs and ground vehicles, and pursuit-evasion games in which a team of UAVs and ground vehicles pursue a second team of evaders while concurrently building a map in an unknown environment (Vidal et al., 2002b).

The cooperation between aerial and ground robots is also researched at University of Southern California. In (Sukhatme et al., 2001), different cooperation cases are studied, such as the use of an aerial robot in a marsupial-style deployment of a small wheeled robot and the localisation of an aerial robot by visually locating and communicating with a ground robot. “Micro” air vehicles (MAV) are also researched in this framework (Vaughan et al., 2000). Furthermore, the Raptor project (Saripalli et al., 2002) considers the use of small electric-powered radio-controlled model helicopters with micro-controllers and Micro Electro-Mechanical based Sensors (MEMS). Relative localization of each robot is accomplished using only local sensing (CMOS camera), in contrast to global localization techniques (GPS), and then it could be applied in environments where GPS is not available (i.e. indoors or between skyscrapers).

[7] http://acl.mit.edu/

2.8.3 Multi-UAV cooperative perception

The application of cooperative perception for teams of UAVs are more rare than in teams of ground robots.

In the BEAR project, pursuit-evasion games considering UAVs and Unmanned Ground Vehicles (UGV) are considered. A probabilistic framework is employed for obstacle and evaders position estimation by the fleet of pursuers (UAVs), and to determine pursuit polices (Vidal et al., 2002b). In this case, non-parametric state representations are employed (grids), but the data fusion and policy determination is carried out by a central node.

Closest to part of the work presented in this thesis is the work presented in (Sukkarieh et al., 2003). In the ANSER project described there, an architecture for multi-vehicle data fusion is designed, and its application to multi-UAV SLAM using vision is presented. State estimation is performed distributelly using the information form of the Kalman filter. Each vehicle uses the information received to update its state and its local map. The map information (represented as the location of a discrete set of landmarks) estimated for each vehicle is propagated to the rest of the fleet. In this work, artificial landmarks of known size are used in order to obtain range and bearing measurements. However, the main issues regarding decentralized data fusion are considered.

In the context of the same group, Grocholsky et al. (2003) present techniques for coordination of sensor platforms in order to maximize the information gain, which is important for cooperative perception activities. It describes results derived from the information form of the Kalman filter or Information Filter. The use of non-parametric belief representations for cooperative perception has been less covered. A study on the use of particle filters on the same context has been presented recently (Ong et al., 2005b).

In (Flint et al., 2004) evidence grids are employed for multi-UAV search missions. The UAVs are able to share environment information. They illustrate how it is possible to achieve some passive coordination due to the fact that the robots share information. Only results in simulation are shown.

Ousingsawat and Campbell (2004) have presented simulation results about the coordination of several UAVs in tracking applications. The position of the UAVs is selected so that it maximizes the information gain. The Fisher Information Matrix of Gaussian estimations is used as a measure of this information gain.

In (Yanli et al., 2005), the authors present a multi-UAV map-building approach based on evidential reasoning. The objective is that a group of UAVs builds a certainty grid about potential targets on an area. The paper shows results only in simulation, and it is very interesting as it presents a comparison with a Bayesian approach to the same problem. The authors conclude that the Dempster-Shafer approach can yield better results in terms of timing when the sensors' accuracy is low. Also, the paper considers cooperative path planning methods based on the results from both, the Bayesian and evidential approaches. Nevertheless, the algorithm presented is purely centralized, and nothing is said about a decentralized version for the evidential approach. Also, (Yu et al., 2004) presents an analysis in simulation of evidential approaches for airborne recognition in military applications.

In (Bertuccelli and How, 2006), evidence grids are used for multi-UAV exploration in the search of

targets. Beta probabilities are used to encode uncertain information about prior knowledge provided by mission designers. The objective is to devise optimal actions for reducing the uncertainty in exploration missions. Results in simulation are shown, but the topics of information fusion are not covered.

I believe in evidence. I believe in observation, measurement, and reasoning, confirmed by independent observers

Isaac Asimov
The Roving Mind (1983)

Chapter 3

Distributed cooperative perception employing information filters

The previous chapter presented the problem of probabilistic cooperative perception in a broad sense, with little details about the actual implementation and applications. In general, however, the fundamental equations presented above are not affordable without further assumptions about the particular problem and the distributions employed to represent the beliefs, and the prediction and measurement functions considered.

One of the most used belief representations are Gaussian distributions, which can be employed to represent unimodal beliefs, with a value with maximum probability (the mean) and uncertainty represented by the covariance of the distribution. They are used elsewhere to represent beliefs about the pose of robots in the localization problem, positions of objects and landmarks in mapping activities, etc.

The chapter presents how a fleet of robots can share and fuse information when it is represented in this way. Moreover, the ideas are applied to the problem of the detection and localization of events of interest by a team of UAVs. As said in Chapter 1, this is one important and valuable application for a team of UAVs. The limited resolution of the sensors on board the vehicles always introduces errors on the estimated localization of an event of interest. It will be shown how the presence of several vehicles can be used to improve the localization estimates.

3.1 Gaussian linear systems: Kalman Filters and Information Filters

Equation (2.7) is used locally by each robot to estimate its belief state. In general, this equation is hard to compute, and no analytical solution is available. However, there is a set of situations in which there is a close solution to the Bayes filter.

One of these situations is the so-called Gaussian Filter. In this filter, the belief state is represented by a multivariate Gaussian distribution, determined by eq. (3.1) for the N-dimensional case, and which is characterized by two parameters, the mean $\boldsymbol{\mu}_t$ and the covariance matrix $\boldsymbol{\Sigma}_t$. The main

properties of this distribution are depicted in Appendix A.

Under certain circumstances, the Bayes filter for updating beliefs expressed using Gaussian representations has an analytical solution, the Kalman Filter (Kalman, 1960). The filter will be briefly presented for completeness and, more importantly, to introduce the dual of the Kalman filter, the Information Filter, that has some implications in multi-robot data fusion applications.

$$bel(\mathbf{X}_t) = \frac{1}{(\sqrt{2\pi})^N \sqrt{|\mathbf{\Sigma}_t|}} e^{-\frac{1}{2}(\mathbf{X}_t - \boldsymbol{\mu}_t)^T \mathbf{\Sigma}_t^{-1} (\mathbf{X}_t - \boldsymbol{\mu}_t)} \tag{3.1}$$

3.1.1 The Kalman Filter

The Kalman Filter (KF) is one of the best known tools for state estimation and filtering. Besides the Markovian assumption, the Kalman filter is a solution of the general Bayes filter if the following restrictions are met (Stone et al., 1999):

1. The prediction step should produce a Gaussian distribution when applied to a priori Gaussian distribution. For the discrete time case, this is equivalent to have the state at time t being a linear function of the state at time $t-1$, plus additive Gaussian noise[1]:

$$\mathbf{X}_t = \mathbf{A}_t \mathbf{X}_{t-1} + \boldsymbol{\nu}_t, \tag{3.2}$$

   where

$$\boldsymbol{\nu}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t). \tag{3.3}$$

   Considering the properties of the Gaussian distributions, in this case, after the prediction step:

$$p(\mathbf{X}_t | \mathbf{X}_{t-1}) \sim \mathcal{N}(\mathbf{A}_t \mathbf{X}_{t-1}, \mathbf{R}_t). \tag{3.4}$$

2. The measurements $\mathbf{Z}_t$ are linear functions of the state a time t plus Gaussian noise.

$$\mathbf{Z}_t = \mathbf{M}_t \mathbf{X}_t + \boldsymbol{\varepsilon}_t, \tag{3.5}$$

   where $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_t)$. In this case, the measurement model is:

$$p(\mathbf{Z}_t | \mathbf{X}_t) \sim \mathcal{N}(\mathbf{M}_t \mathbf{X}_t, \mathbf{S}_t). \tag{3.6}$$

3. The initial prior $p(\mathbf{X}_0)$ should be Gaussian, with mean $\boldsymbol{\mu}_0$ and covariance matrix $\mathbf{\Sigma}_0$.

[1] From now on, unless otherwise indicated, it is removed the explicit dependence of the prediction on the robot actions $\mathbf{u}_t$, as most of the thesis deals with states that are not dependant on robot actions.

Algorithm 3.1 $(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t) \leftarrow$ Kalman Filter$(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1}, \mathbf{z}_t)$

1: $\bar{\boldsymbol{\Sigma}}_t = \mathbf{A}_t \boldsymbol{\Sigma}_{t-1} \mathbf{A}_t^T + \mathbf{R}_t$ /* *Prediction* */
2: $\bar{\boldsymbol{\mu}}_t = \mathbf{A}_t \boldsymbol{\mu}_{t-1}$
3: $p(\mathbf{X}_t|\mathbf{z}^{t-1}) \sim \mathcal{N}(\bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)$
4: $\mathbf{K}_t = \bar{\boldsymbol{\Sigma}}_t \mathbf{M}_t (\mathbf{M}_t \bar{\boldsymbol{\Sigma}}_t \mathbf{M}_t^T + \mathbf{S}_t)^{-1}$
5: $\boldsymbol{\mu}_t = \bar{\boldsymbol{\mu}}_t + \mathbf{K}_t(\mathbf{z}_t - \mathbf{M}_t \bar{\boldsymbol{\mu}}_t)$/* *Update* */
6: $\boldsymbol{\Sigma}_t = (\mathbf{I} - \mathbf{K}_t \mathbf{M}_t) \bar{\boldsymbol{\Sigma}}_t$
7: $p(\mathbf{X}_t|\mathbf{z}^t) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$

If these circumstances hold, the posterior $p(\mathbf{X}_t|\mathbf{z}^t)$ is always Gaussian, for any time instant t, that is, the Gaussian distribution is its own *conjugate-prior* under a Gaussian likelihood. Conjugate-priors are interesting distributions because they lead to efficient solutions of the Bayes filter.

The filtering distribution (2.7) is then solved recursively by using Algorithm 3.1, which obtains the current mean and covariance matrix $\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t$ given the previous estimation $bel(\mathbf{X}_{t-1}) \sim \mathcal{N}(\boldsymbol{\mu}_{t-1}, \boldsymbol{\Sigma}_{t-1})$ and the actual current measurements $\mathbf{z}_t$.

3.1.2 The Information Filter

There is a dual implementation of the Kalman filter, the so called the Information Filter (IF). The restrictions for the application of the IF are the same as in the KF. The main difference comes from the way the Gaussian distributions are represented. While the KF represents the distribution using its first and second order moments, the IF employs the so-called *canonical representation*. The fundamental elements are the *information vector* $\boldsymbol{\xi}$ and the *information matrix* $\boldsymbol{\Omega}$, defined as:

$$\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1} \tag{3.7}$$

$$\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1} \mu \tag{3.8}$$

Using these elements, the probability density function of a multivariate Gaussian distribution can be expressed as:

$$bel(\mathbf{X}_t) = \eta e^{-\frac{1}{2}\mathbf{X}_t^T \boldsymbol{\Omega}_t \mathbf{X}_t + \mathbf{X}_t^T \boldsymbol{\xi}_t} \tag{3.9}$$

The prediction and updating equations for the IF can be derived from the usual KF. Firstly, the predicted information matrix $\bar{\boldsymbol{\Omega}}_t$ (step 1 of Algorithm 3.1):

$$\bar{\boldsymbol{\Omega}}_t = \bar{\boldsymbol{\Sigma}}_t^{-1} = (\mathbf{A}_t \boldsymbol{\Omega}_{t-1}^{-1} \mathbf{A}_t^T + \mathbf{R}_t)^{-1} \tag{3.10}$$

Defining

$$\boldsymbol{\Psi}_t = (\mathbf{A}_t \boldsymbol{\Omega}_{t-1}^{-1} \mathbf{A}_t^T)^{-1} = \mathbf{A}_t^{-T} \boldsymbol{\Omega}_{t-1} \mathbf{A}_t^{-1} \tag{3.11}$$

and using the matrix inversion lemma[2]:

[2] $(\mathbf{A} + \mathbf{X}\mathbf{B}\mathbf{X}^T)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{X}(\mathbf{B}^{-1} + \mathbf{X}^T\mathbf{A}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{A}^{-1}$

Algorithm 3.2 $(\boldsymbol{\xi}_t, \boldsymbol{\Omega}_t) \leftarrow$ Information Filter$(\boldsymbol{\xi}_{t-1}, \boldsymbol{\Omega}_{t-1}, \mathbf{z}_t)$

1: $\boldsymbol{\Psi}_t = \mathbf{A}_t^{-T} \boldsymbol{\Omega}_{t-1} \mathbf{A}_t^{-1}$
2: $\bar{\boldsymbol{\Omega}}_t = \boldsymbol{\Psi}_t - \boldsymbol{\Psi}_t (\mathbf{R}_t^{-1} + \boldsymbol{\Psi}_t)^{-1} \boldsymbol{\Psi}_t$
3: $\bar{\boldsymbol{\xi}}_t = \bar{\boldsymbol{\Omega}}_t \mathbf{A}_t \boldsymbol{\Omega}_{t-1}^{-1} \boldsymbol{\xi}_{t-1}$
4: $p(\mathbf{X}_t | \mathbf{z}^{t-1}) \sim \mathcal{N}(\bar{\boldsymbol{\Omega}}_t^{-1} \bar{\boldsymbol{\xi}}_t, \bar{\boldsymbol{\Omega}}_t^{-1})$
5: $\boldsymbol{\Omega}_t = \bar{\boldsymbol{\Omega}}_t + \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t$
6: $\boldsymbol{\xi}_t = \bar{\boldsymbol{\xi}}_t + \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{z}_t$
7: $p(\mathbf{X}_t | \mathbf{z}^t) \sim \mathcal{N}(\boldsymbol{\Omega}_t^{-1} \boldsymbol{\xi}_t, \boldsymbol{\Omega}_t^{-1})$

$$\bar{\boldsymbol{\Omega}}_t = (\boldsymbol{\Psi}_t^{-1} + \mathbf{R}_t)^{-1} = \boldsymbol{\Psi}_t - \boldsymbol{\Psi}_t (\mathbf{R}_t^{-1} + \boldsymbol{\Psi}_t)^{-1} \boldsymbol{\Psi}_t \tag{3.12}$$

Also, the predicted information vector $\bar{\boldsymbol{\mu}}_t$ is computed as follows:

$$\bar{\boldsymbol{\mu}}_t = \mathbf{A}_t \boldsymbol{\mu}_{t-1} \Rightarrow \bar{\boldsymbol{\Omega}}_t^{-1} \bar{\boldsymbol{\xi}}_t = \mathbf{A}_t \boldsymbol{\Omega}_{t-1}^{-1} \boldsymbol{\xi}_{t-1} \Rightarrow \bar{\boldsymbol{\xi}}_t = \bar{\boldsymbol{\Omega}}_t \mathbf{A}_t \boldsymbol{\Omega}_{t-1}^{-1} \boldsymbol{\xi}_{t-1} \tag{3.13}$$

The update step can be computed from the original Bayes updating step (2.7):

$$p(\mathbf{X}_t | \mathbf{z}^t) = \eta \underbrace{p(\mathbf{z}_t | \mathbf{X}_t)}_{\sim N(\mathbf{M}_t \mathbf{X}_t, \mathbf{S}_t)} \underbrace{p(\mathbf{X}_t | \mathbf{z}^{t-1})}_{\sim N(\bar{\boldsymbol{\mu}}_t, \bar{\boldsymbol{\Sigma}}_t)} \tag{3.14}$$

Then

$$p(\mathbf{X}_t | \mathbf{z}^t) = \eta \exp\{-0.5(\mathbf{z}_t - \mathbf{M}_t \mathbf{X}_t)^T \mathbf{S}_t^{-1} (\mathbf{z}_t - \mathbf{M}_t \mathbf{X}_t) - 0.5(\mathbf{X}_t - \bar{\boldsymbol{\mu}}_t)^T \bar{\boldsymbol{\Sigma}}_t^{-1} (\mathbf{X}_t - \bar{\boldsymbol{\mu}}_t)\} \tag{3.15}$$

considering that $\bar{\boldsymbol{\Sigma}}_t^{-1} = \bar{\boldsymbol{\Omega}}_t$ and reordering terms:

$$p(\mathbf{X}_t | \mathbf{z}^t) = \eta \exp\{-0.5 \mathbf{X}_t^T (\mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t + \bar{\boldsymbol{\Omega}}_t) \mathbf{X}_t + \mathbf{X}_t^T (\mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{z}_t + \bar{\boldsymbol{\xi}}_t)\} \tag{3.16}$$

Comparing (3.16) with (3.9), then:

$$\boldsymbol{\Omega}_t = \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t + \bar{\boldsymbol{\Omega}}_t \tag{3.17}$$

$$\boldsymbol{\xi}_t = \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{z}_t + \bar{\boldsymbol{\xi}}_t \tag{3.18}$$

The dual Information Filter for the Kalman filter of Algorithm 3.1 is given by Algorithm 3.2.

The IF presents some advantages and drawbacks when compared to the KF. One of the advantages of the canonical representation for the IF is that it can consider complete uncertainty seamlessly in the filter, by setting $\boldsymbol{\Omega}_t = \mathbf{0}$. The prediction and updating steps are dual in the KF and IF, in the sense that the prediction is more complicated in the IF than in the KF, but, on the other hand, the update steps are easier (Appendix A). Moreover, the additive nature of the updating steps 5 and 6 of Algorithm 3.2 is what makes the information filter interesting for multi-robot applications.

3.1.3 Information filter for the full state trajectory

The information form also presents an interesting property when the full state trajectory $bel(\mathbf{X}^t)$ is considered. If the assumptions for the Information Filter hold, it can be seen that the joint distribution over the full state is also Gaussian. The IF for the case of the full joint distribution can be derived from the general equation:

$$bel(\mathbf{X}^t) = p(\mathbf{X}^t|\mathbf{z}^t) = \eta^{'} p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} p(\mathbf{z}_\tau|\mathbf{X}_\tau) p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1}) \tag{3.19}$$

If the prior is Gaussian, then $bel(\mathbf{X}^0) = p(\mathbf{X}_0) = \eta e^{-\frac{1}{2}\mathbf{X}_0^T \mathbf{\Omega}_0 \mathbf{X}_0 + \mathbf{X}_0^T \boldsymbol{\xi}_0}$. The belief state up to time 1 is given by $bel(\mathbf{X}^1) = \eta^{'} p(\mathbf{z}_1|\mathbf{X}_1) p(\mathbf{X}_1|\mathbf{X}_0) bel(\mathbf{X}^0)$. In logarithmic form:

$$\log bel(\mathbf{X}^t) = C + \log p(\mathbf{X}_0) + \log p(\mathbf{z}_1|\mathbf{X}_1) + \log p(\mathbf{X}_1|\mathbf{X}_0) \tag{3.20}$$

where $C = \log \eta^{'}$ is a normalizing constant. Following equations (3.2) and (3.5), the prediction and measurement distributions are also Gaussians, $p(\mathbf{X}_1|\mathbf{X}_0) = \mathcal{N}(\mathbf{A}_1\mathbf{X}_0, \mathbf{R}_1)$ and $p(\mathbf{Z}_1|\mathbf{X}_1) = \mathcal{N}(\mathbf{M}_1\mathbf{X}_1, \mathbf{S}_1)$. Therefore:

$$\begin{aligned} \log bel(\mathbf{X}^1) = C - \frac{1}{2}\mathbf{X}_0^T \mathbf{\Omega}_0 \mathbf{X}_0 + \mathbf{X}_0^T \boldsymbol{\xi}_0 - \underbrace{\frac{1}{2}[(\mathbf{X}_1 - \mathbf{A}_1\mathbf{X}_0)^T \mathbf{R}_1^{-1} (\mathbf{X}_1 - \mathbf{A}_1\mathbf{X}_0)]}_{prediction} - \\ - \underbrace{\frac{1}{2}[(\mathbf{z}_1 - \mathbf{M}_1\mathbf{X}_1)^T \mathbf{S}_1^{-1} (\mathbf{z}_1 - \mathbf{M}_1\mathbf{X}_1)]}_{update} \end{aligned} \tag{3.21}$$

The state up to time 1 is represented by the vector $\mathbf{X}^1 = \begin{pmatrix} \mathbf{X}_1^T & \mathbf{X}_0^T \end{pmatrix}^T$. Then $\mathbf{X}_1 - \mathbf{A}_1\mathbf{X}_0 = \begin{pmatrix} \mathbf{I} & -\mathbf{A}_1 \end{pmatrix} \mathbf{X}^1$. Reordering terms:

$$\begin{aligned} \log bel(\mathbf{X}^1) &= C - \frac{1}{2}\big[\mathbf{X}_0^T \mathbf{\Omega}_0 \mathbf{X}_0 - \mathbf{X}_0^T \boldsymbol{\xi}_0 + \\ &\quad + \mathbf{X}^{1T} \begin{pmatrix} \mathbf{I} - \mathbf{A}_1^T \end{pmatrix} \mathbf{R}_1^{-1} \begin{pmatrix} \mathbf{I} & -\mathbf{A}_1 \end{pmatrix} \mathbf{X}^1 + (\mathbf{z}_1 - \mathbf{M}_1\mathbf{X}_1)^T \mathbf{S}_1^{-1} (\mathbf{z}_1 - \mathbf{M}_1\mathbf{X}_1)\big] = \\ &= C' - \frac{1}{2}\mathbf{X}^{1T}\big[\underbrace{\begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{\Omega}_0 \end{pmatrix}}_{prior} + \underbrace{\begin{pmatrix} \mathbf{I} \\ -\mathbf{A}_1^T \end{pmatrix} \mathbf{R}_1^{-1} \begin{pmatrix} \mathbf{I} & -\mathbf{A}_1 \end{pmatrix}}_{prediction} + \underbrace{\begin{pmatrix} \mathbf{M}_1^T \mathbf{S}_1^{-1} \mathbf{M}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix}}_{update}\big]\mathbf{X}^1 - \\ &\quad - \mathbf{X}^{1T}\big[\underbrace{\begin{pmatrix} \mathbf{0} \\ \boldsymbol{\xi}_0 \end{pmatrix}}_{prior+prediction} + \underbrace{\begin{pmatrix} \mathbf{M}_1^T \mathbf{S}_1^{-1} \mathbf{z}_1 \\ \mathbf{0} \end{pmatrix}}_{update}\big] \end{aligned} \tag{3.22}$$

Comparing this expression again with the canonical form of a multivariate Gaussian distribution (in logarithmic form), it can be seen that the result is also Gaussian. In the prediction step, the information matrix and vector are augmented:

Algorithm 3.3 $(\boldsymbol{\xi}^t, \boldsymbol{\Omega}^t) \leftarrow$Information Filter$(\boldsymbol{\xi}^{t-1}, \boldsymbol{\Omega}^{t-1}, \mathbf{z}_t)$

1: $\bar{\boldsymbol{\Omega}}^t = \textbf{Augment\_Matrix}(\ \boldsymbol{\Omega}^{t-1}) + \begin{pmatrix} \begin{pmatrix} \mathbf{I} \\ -\mathbf{A}_t^T \end{pmatrix} \mathbf{R}_t^{-1} \begin{pmatrix} \mathbf{I} & -\mathbf{A}_t \end{pmatrix} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$

2: $\bar{\boldsymbol{\xi}}^t = \textbf{Augment\_Vector}\ (\boldsymbol{\xi}^{t-1})$

3: $\boldsymbol{\Omega}^t = \bar{\boldsymbol{\Omega}}^t + \begin{pmatrix} \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix}$

4: $\boldsymbol{\xi}^t = \bar{\boldsymbol{\xi}}_t + \begin{pmatrix} \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{z}_t \\ \mathbf{0} \end{pmatrix}$

$$\bar{\boldsymbol{\Omega}}^1 = \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Omega}_0 \end{pmatrix} + \begin{pmatrix} \mathbf{R}_t^{-1} & -\mathbf{R}_t^{-1}\mathbf{A}_t \\ -\mathbf{A}_t^T\mathbf{R}_t^{-1} & \mathbf{A}_t^T\mathbf{R}_t^{-1}\mathbf{A}_t \end{pmatrix} \tag{3.23}$$

$$\bar{\boldsymbol{\xi}}^1 = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\xi}_0 \end{pmatrix} \tag{3.24}$$

If a new measurement is received, the predicted information vector and matrix are updated :

$$\boldsymbol{\Omega}^1 = \bar{\boldsymbol{\Omega}}^1 + \begin{pmatrix} \mathbf{M}_1^T \mathbf{S}_1^{-1} \mathbf{M}_1 & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{3.25}$$

$$\boldsymbol{\xi}^1 = \bar{\boldsymbol{\xi}}^1 + \begin{pmatrix} \mathbf{M}_1^T \mathbf{S}_1^{-1} \mathbf{z}_1 \\ \mathbf{0} \end{pmatrix} \tag{3.26}$$

The same procedure can be applied recursively up to time t. Thus, knowing the information matrix and vector up to time $t-1$, $\boldsymbol{\Omega}^{t-1}$ and $\boldsymbol{\xi}^{t-1}$, the prediction steps are:

$$\bar{\boldsymbol{\Omega}}^t = \begin{pmatrix} \mathbf{0} & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \boldsymbol{\Omega}_{(t-1)(t-1)} & \cdots \\ \mathbf{0} & \vdots & \ddots \end{pmatrix} + \begin{pmatrix} \mathbf{R}_t^{-1} & -\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ -\mathbf{A}_t^T\mathbf{R}_t^{-1} & \mathbf{A}_t^T\mathbf{R}_t^{-1}\mathbf{A}_t & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{3.27}$$

$$\bar{\boldsymbol{\xi}}^t = \begin{pmatrix} \mathbf{0} \\ \boldsymbol{\xi}^{t-1} \end{pmatrix} \tag{3.28}$$

And, if one measurement is received, the updating equations are:

$$\boldsymbol{\Omega}^t = \bar{\boldsymbol{\Omega}}^t + \begin{pmatrix} \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t & \mathbf{0}^T & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{3.29}$$

$$\boldsymbol{\xi}^t = \bar{\boldsymbol{\xi}}^t + \begin{pmatrix} \mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{z}_t \\ \mathbf{0} \end{pmatrix} \tag{3.30}$$

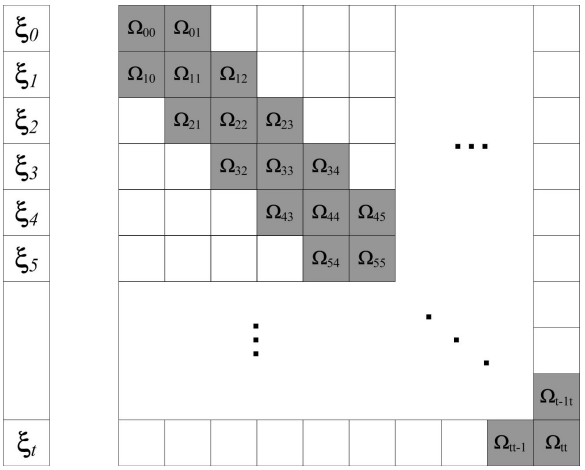

Figure 3.1: Structure of the information matrix for the full trajectory. The information matrix is a block tridiagonal symmetric matrix, due to the Markov structure of the process.

The Information Filter for estimating the belief over the state trajectory is summarized in Algorithm 3.3, where **Augment_Matrix** adds a block row and a block column to the previous information matrix and **Augment_Vector** adds a block to the previous information vector.

Evidently, the state grows along time. In the general case of an information matrix, for a N-dimensional state, the storage required is $O(N^2)$. However, in this case, as it can be seen from the prediction and updating equations, the matrix structure is block tridiagonal and symmetric (see Fig. 3.1) at any time, and thus the storage required is $O(N)$ (where N is the number of time steps). Also, the computational complexity of the algorithm itself is $O(1)$, as the prediction and updating computations at each time instant only involve the previous block. In the case of the Kalman filter, the corresponding matrix is, however, fully correlated.

State reduction

In certain situations, the length of the trajectory estimated should be limited, for instance because of storage or bandwidth restrictions. Therefore, a method for reducing the state whenever the size of the trajectory grows over a given threshold is required.

In order to do so, the removed part of the trajectory should be marginalized out. Consider first the case of eliminating the information at time 0. Representing by $\mathbf{X}^{k:t}$ the state trajectory between times k and t, eliminating the information corresponding to the state at time 0 consists of computing $bel(\mathbf{X}^{1:t}) = \int bel(\mathbf{X}^{0:t}) d\mathbf{X}_0$. As seen in Appendix A, the marginal of a multivariate Gaussian in the canonical form can be computed in closed form. Moreover, due to the structure of the information matrix for this case, the computations required only involve local block matrix operations (see Fig. 3.2). For instance, eliminating the information about the state at time 0 only affects the blocks corresponding to the state at time 1 $\mathbf{\Omega}_{11}$ and $\boldsymbol{\xi}_{11}$, which become:

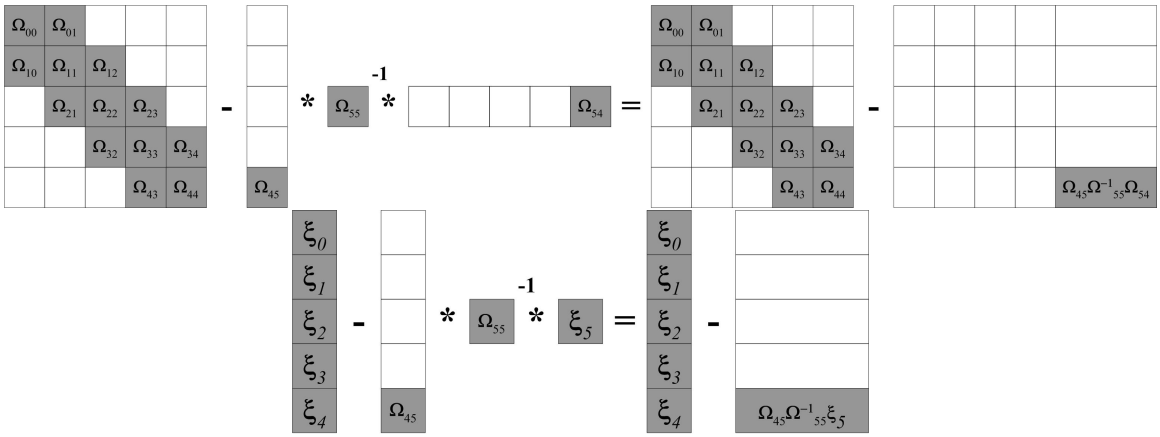

Figure 3.2: Marginalization of the removed point of the trajectory. Due to the structure of the information matrix, the marginalization only involves local block operations.

$$\boldsymbol{\Omega}_{11} \leftarrow \boldsymbol{\Omega}_{11} - \boldsymbol{\Omega}_{10}\boldsymbol{\Omega}_{00}^{-1}\boldsymbol{\Omega}_{01} \tag{3.31}$$

$$\boldsymbol{\xi}_{1} \leftarrow \boldsymbol{\xi}_{1} - \boldsymbol{\Omega}_{10}\boldsymbol{\Omega}_{00}^{-1}\boldsymbol{\xi}_{0} \tag{3.32}$$

Moreover, this marginalization operation maintains the block tridiagonal structure of the matrix. It can be easily seen that eliminating an intermediate state also maintains the structure of the matrix. In general, if the information at time t is eliminated, the only blocks affected are those linked to it (that is, $t-1$ and $t+1$), following:

$$\begin{aligned}
\boldsymbol{\Omega}_{t-1t-1} &\leftarrow \boldsymbol{\Omega}_{t-1t-1} - \boldsymbol{\Omega}_{tt-1}^{T}\boldsymbol{\Omega}_{tt}^{-1}\boldsymbol{\Omega}_{tt-1} \\
\boldsymbol{\xi}_{t-1} &\leftarrow \boldsymbol{\xi}_{t-1} - \boldsymbol{\Omega}_{tt-1}^{T}\boldsymbol{\Omega}_{tt}^{-1}\boldsymbol{\xi}_{t} \\
\boldsymbol{\Omega}_{t+1t+1} &\leftarrow \boldsymbol{\Omega}_{t+1t+1} - \boldsymbol{\Omega}_{t+1t}\boldsymbol{\Omega}_{tt}^{-1}\boldsymbol{\Omega}_{t+1t}^{T} \\
\boldsymbol{\xi}_{t+1} &= \boldsymbol{\xi}_{t+1} - \boldsymbol{\Omega}_{t+1t}^{T}\boldsymbol{\Omega}_{tt}^{-1}\boldsymbol{\xi}_{t} \\
\boldsymbol{\Omega}_{t+1t-1} &\leftarrow -\boldsymbol{\Omega}_{t+1t}\boldsymbol{\Omega}_{tt}^{-1}\boldsymbol{\Omega}_{tt-1}
\end{aligned} \tag{3.33}$$

3.2 Multi-robot perception using Information Filters

The Information Filter allows implementing the expressions described in Chapter 2 quite directly. For instance, it is straightforward to determine the Information Filter for the centralized multi-robot perception case.

Recalling (2.11), the updating step is given by the product of the individual likelihoods for the data of all the robots $\prod_{j=1}^{M} p(\mathbf{z}_{j,t}|\mathbf{X}_t)$. If the assumptions of the IF hold, and as the likelihood

functions $p(\mathbf{z}_{j,t}|\mathbf{X}_t)$ are Gaussian, then this updating step consist of a sum of all the information contributions from all the robots of the fleet.

$$\boldsymbol{\Omega}_t = \bar{\boldsymbol{\Omega}}_t + \sum_{j=1}^{M} \mathbf{M}_{j,t}^T \mathbf{S}_{j,t}^{-1} \mathbf{M}_{j,t} \tag{3.34}$$

$$\boldsymbol{\xi}_t = \bar{\boldsymbol{\xi}}_t + \sum_{j=1}^{M} \mathbf{M}_{j,t}^T \mathbf{S}_{j,t}^{-1} \mathbf{z}_{j,t} \tag{3.35}$$

Therefore, in the case a centralized node receives all the measurements from the robots, these expressions can be used to obtain the global state update.

3.2.1 Decentralized Information Filter for the static case

The IF leads to a decentralized implementation in which each UAV maintains a local belief employing only local information, and then shares this belief with the local neighbors. The case of an static state is considered first.

In this case, each UAV will run locally Algorithm 3.2, with $\mathbf{A}_t = \mathbf{I}$ and $\mathbf{R}_t = \mathbf{0}$ (and therefore, steps 1 to 3 do not apply in this case). And, for updating the status of its alarms, it will use the information obtained from its sensors. Of course, the estimated belief state will be different from the centralized one due to the missed terms in the updating steps 5 and 6 (the information collected by the rest of the robots). The information vector $\boldsymbol{\xi}_t^i$ and information matrix $\boldsymbol{\Omega}_t^i$ represents the local belief state of UAV i at time t.

When UAV i flies within communication range with other UAV j, they can share their beliefs. UAV i can apply the relation (2.20) to update its belief state. It is easy to see (from the logarithmic form of that equation, and considering the form of the Gaussian distribution in canonical form) that the resultant equations for the updating of the information vector and matrix are very similar to the update steps 5 and 6 of Algorithm 3.2:

$$\begin{aligned} \log bel_i(\mathbf{X}_t) \leftarrow \log(C) + \log bel_i(\mathbf{X}_t) + \log bel_j(\mathbf{X}_t) - \log bel_j(\mathbf{X}_{t-1}) = \\ = \log(C) - \frac{1}{2}\mathbf{X}^T\boldsymbol{\Omega}_t^i\mathbf{X} + \mathbf{X}^T\boldsymbol{\xi}_t^i - \frac{1}{2}\mathbf{X}^T\boldsymbol{\Omega}_t^j\mathbf{X} + \mathbf{X}^T\boldsymbol{\xi}_t^j - \frac{1}{2}\mathbf{X}^T\boldsymbol{\Omega}_{t-1}^j\mathbf{X} + \mathbf{X}^T\boldsymbol{\xi}_{t-1}^j = \\ = \log(C) - \frac{1}{2}\big[\mathbf{X}^T(\boldsymbol{\Omega}_t^i + \boldsymbol{\Omega}_t^j - \boldsymbol{\Omega}_{t-1}^j)\mathbf{X}\big] + \mathbf{X}^T(\boldsymbol{\xi}_t^i + \boldsymbol{\xi}_t^j - \boldsymbol{\xi}_{t-1}^j) \end{aligned} \tag{3.36}$$

Comparing this with (3.9), the resultant information vector and matrix are incremented with the increase in evidence $\boldsymbol{\Omega}_t^i \leftarrow \boldsymbol{\Omega}_t^i + (\boldsymbol{\Omega}_t^j - \boldsymbol{\Omega}_{t-1}^j)$ and $\boldsymbol{\xi}_t^i \leftarrow \boldsymbol{\xi}_t^i + \boldsymbol{\xi}_t^j - \boldsymbol{\xi}_{t-1}^j$, where $\boldsymbol{\Omega}_{t-1}^j, \boldsymbol{\xi}_{t-1}^j$ in this case denotes the last received belief from UAV j (whenever it was received).

Again, the previous relations only hold if UAV i is just a sink of information. If UAV i has previously communicated with UAV j, then it is needed to remove also the common information shared by both UAVs. This common information is also a Gaussian distribution and that is represented by $\boldsymbol{\xi}_t^{ij}$ and $\boldsymbol{\Omega}_t^{ij}$.

Algorithm 3.4 Decentralized_Information_Filter(i)

1: $\boldsymbol{\Omega}_0^i = \mathbf{0}$/* *Prior* */
2: $\boldsymbol{\xi}_0^i = \mathbf{0}$
3: **while** true **do**
4: **if** New data $\mathbf{z}_{i,t}$ **then** /* *Local information* */
5: $\boldsymbol{\Omega}_t^i = \boldsymbol{\Omega}_{t-1}^i + \mathbf{M}_{i,t}^T \mathbf{S}_{i,t}^{-1} \mathbf{M}_{i,t}$
6: $\boldsymbol{\xi}_t^i = \boldsymbol{\xi}_{t-1}^i + \mathbf{M}_{i,t}^T \mathbf{S}_{i,t}^{-1} \mathbf{z}_{i,t}$
7: **end if**
8: **if** New belief from UAV j **then**
9: $\boldsymbol{\Omega}_t^i \leftarrow \boldsymbol{\Omega}_t^i + (\boldsymbol{\Omega}_t^j - \boldsymbol{\Omega}_t^{ij})$/* *Remove common info* */
10: $\boldsymbol{\xi}_t^i \leftarrow \boldsymbol{\xi}_t^i + (\boldsymbol{\xi}_t^j - \boldsymbol{\xi}_t^{ij})$
11: $\boldsymbol{\Omega}_t^{ij} = \boldsymbol{\Omega}_t^j$
12: $\boldsymbol{\xi}_t^{ij} = \boldsymbol{\xi}_t^j$
13: **end if**
14: **if** Communication event **then**
15: **for all** UAV j within comm. range **do**
16: Send belief to UAV j
17: $\boldsymbol{\Omega}_t^{ij} = \boldsymbol{\Omega}_t^i$
18: $\boldsymbol{\xi}_t^{ij} = \boldsymbol{\xi}_t^i$
19: **end for**
20: **end if**
21: **end while**

Any time that information is received from UAV j, after the fusion step this common information is updated as:

$$\begin{aligned} \boldsymbol{\Omega}_t^{ij} &= \boldsymbol{\Omega}_t^j \\ \boldsymbol{\xi}_t^{ij} &= \boldsymbol{\xi}_t^j \end{aligned} \tag{3.37}$$

while when the UAV i sends information to UAV j, the common information is set to (Nettleton et al., 2003):

$$\begin{aligned} \boldsymbol{\Omega}_t^{ij} &= \boldsymbol{\Omega}_t^i \\ \boldsymbol{\xi}_t^{ij} &= \boldsymbol{\xi}_t^i \end{aligned} \tag{3.38}$$

If there are more than one UAV within communication range, then, if $C(i)$ is the set of neighbors of UAV i:

$$\boldsymbol{\Omega}_t^i \leftarrow \boldsymbol{\Omega}_t^i + \sum_{j \in C(i)} (\boldsymbol{\Omega}_t^j - \boldsymbol{\Omega}_t^{ij}) \tag{3.39}$$

$$\boldsymbol{\xi}_t^i \leftarrow \boldsymbol{\xi}_t^i + \sum_{j \in C(i)} (\boldsymbol{\xi}_t^j - \boldsymbol{\xi}_t^{ij}) \tag{3.40}$$

The decentralized filter is then given by Algorithm 3.4. Being the state static, the amount of information required to store the belief state (and to communicate it) is constant along time. Also, the beliefs from other UAVs can be received asynchronously and in any order. Moreover, one UAV does not have to communicate continuously its belief, as it can accumulate evidence and transmit it later (without increasing the storage). Therefore, from the point of view of bandwidth requirements, the transmission of information can be adjusted depending on the network conditions.

3.2.2 Decentralized Information Filter for the dynamic case

Following the reasoning of Section 2.4.2, the formulae for combining the beliefs received by an UAV i from its neighbors is similar to eqs. (3.40), considering the information vector and matrix *over the full trajectory.*

$$\boldsymbol{\Omega}^{t,i} \leftarrow \boldsymbol{\Omega}^{t,i} + \sum_{j \in C(i)} (\boldsymbol{\Omega}^{t,j} - \bar{\boldsymbol{\Omega}}^{t,ij}) \tag{3.41}$$

$$\boldsymbol{\xi}^{t,i} \leftarrow \boldsymbol{\xi}^{t,i} + \sum_{j \in C(i)} (\boldsymbol{\xi}^{t,j} - \bar{\boldsymbol{\xi}}^{t,ij}) \tag{3.42}$$

where $\bar{\boldsymbol{\Omega}}^{t,ij}$ and $\bar{\boldsymbol{\xi}}^{t,ij}$ is the common information shared with UAV j, predicted up to time t. In order to maintain this common information, an additional filter is needed for each communication channel between robots. These filters are implemented following Algorithm 3.3, but without the updating steps 3 and 4. The objective is to relate all the information to the same state space.

In the dynamic case, special care has to be taken considering synchronization issues when combining trajectories. The trajectory is represented at discrete time intervals. The combination formulae will work provided that the differences in these intervals are bounded.

3.2.3 Eliminating unknown common information

The previous equations, as commented in Section 2.5, assume that there are no loops in the network of UAVs, in the sense that the belief information shared between two robots follows a unique path. If it is not the case, prior to combining the beliefs, this unknown common information should be removed. If not, non-consistent estimations could be obtained.

There are several options that can be considered for avoiding an overconfident estimation due to accounting common information several times. The first one is to ensure a tree topology in the belief network (Utete and Durrant-Whyte, 1994), as seen in Section 2.5. However, this option imposes a strong constraint on the potential communication links among the UAVs.

Another option is that each UAV only sends its last local information update at every time stamp (the amount of information obtained by using its sensors, represented by $\mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{M}_t$ and $\mathbf{M}_t^T \mathbf{S}_t^{-1} \mathbf{z}_t$). This way, no information is duplicated. The problem in this case is that the use of an UAV as *data mule* is lost: one UAV that collects the evidence from a group of local neighbors will communicate it to other robots that could be initially disconnected from the firsts. Moreover, if the connection

between two UAVs is lost, it will lose information that would have been available in the case that the robot sends the complete belief.

The last option is to employ a conservative fusion rule, that ensures that the robot does not become overconfident even in presence of duplicated information. For the case of the IF, there is an analytic solution for this, given by the *Covariance Intersection* (CI) algorithm (Julier and Uhlmann, 1997b). The CI algorithm is a way of combining information from random variables whose cross-correlations are unknown.

The CI rule for combining the information matrix and information vector from two Gaussians **A** and **B** (for instance, two estimations from different vehicles of the same quantity) whose cross correlations are not known in order to obtain an estimation of the information matrix and vector of a random variable **C** is:

$$\mathbf{\Omega_c} = \omega\mathbf{\Omega_a} + (1-\omega)\mathbf{\Omega_b} \tag{3.43}$$

$$\boldsymbol{\xi}_\mathbf{c} = \omega\boldsymbol{\xi}_\mathbf{a} + (1-\omega)\boldsymbol{\xi}_\mathbf{b} \tag{3.44}$$

for $\omega \in [01]$. It can be seen that the estimation is consistent in the sense that $\mathbf{\Sigma_c} - \hat{\mathbf{\Sigma}}_\mathbf{c}$ is semi-definite positive (where $\mathbf{\Sigma_c} = \mathbf{\Omega_c}^{-1}$ is the estimated covariance matrix and $\hat{\mathbf{\Sigma}}_\mathbf{c}$ is the actual covariance matrix) for any (unknown) cross-correlation matrix $\mathbf{\Sigma_{ab}}$, and for any ω. The value of ω can be selected following some criteria, as maximizing the obtained determinant of $\mathbf{\Omega_c}$ (minimizing the entropy of the final distribution, see Appendix A). Another option is to use ω as a weight that shows the UAV confidence in its own estimation and the neighbor's ones.

Therefore, the conservative rule to combine the belief of UAV i with that received from UAV j is given by:

$$\mathbf{\Omega}^i \leftarrow \omega\mathbf{\Omega}^i + (1-\omega)\mathbf{\Omega}^j \tag{3.45}$$

$$\boldsymbol{\xi}^i \leftarrow \omega\boldsymbol{\xi}^i + (1-\omega)\boldsymbol{\xi}^j \tag{3.46}$$

These fusion rules, that substitutes lines 9 and 10 of Algorithm 3.4, apply for both, the dynamic and static case. Employing the covariance intersection formula even avoids the need to maintain an estimation of the common information transmitted to the neighbor UAVs. However, as these fusion rules are conservative, some information is lost with respect to the purely centralized case.

3.3 Extensions to non-linear systems

The Kalman and Information Filter are widely used due to their simplicity and limited computational cost. However, in general, the motion equation (3.2) and the measurement equation (3.5) will be non-linear functions on the state.

$$\mathbf{X}_t = \mathbf{f}_t(\mathbf{X}_{t-1}) + \boldsymbol{\nu}_t \tag{3.47}$$

$$\mathbf{Z}_t = \mathbf{g}_t(\mathbf{X}_t) + \boldsymbol{\varepsilon}_t \tag{3.48}$$

In this case, even for a Gaussian *a priori* distribution $p(\mathbf{X}_{t-1}|\mathbf{z}^{t-1})$, the posterior $p(\mathbf{X}_t|\mathbf{z}^t)$ is not Gaussian. Thus, the KF (IF) assumptions do not hold. However, in many cases the resultant distribution is assumed to be Gaussian, because then the number of parameters needed to characterize the distribution is kept low (only the mean and covariance are needed). The transformations of the mean and covariances due to the non-linear prediction and update steps are tackled considering two main options.

3.3.1 First order linearization

One of the approximations for the use of the IF with non-linear systems is to linearize both equations (3.47) and (3.48) around the previous mean $\boldsymbol{\mu}_{t-1}$.

$$\mathbf{X}_t \approx \mathbf{f}_t(\boldsymbol{\mu}_{t-1}) + \nabla\mathbf{f}_t(\boldsymbol{\mu}_{t-1})(\mathbf{X}_{t-1} - \boldsymbol{\mu}_{t-1}) + \boldsymbol{\nu}_t \tag{3.49}$$

$$\mathbf{Z}_t \approx \mathbf{g}_t(\bar{\boldsymbol{\mu}}_t) + \nabla\mathbf{g}_t(\bar{\boldsymbol{\mu}}_t)(\mathbf{X}_t - \bar{\boldsymbol{\mu}}_t) + \boldsymbol{\varepsilon}_t \tag{3.50}$$

with $\bar{\boldsymbol{\mu}}_t = \mathbf{f}_t(\boldsymbol{\mu}_{t-1})$. This leads to the Extended Information Filter (EIF). In the algorithm of the EIF, the matrices $\mathbf{A}_t$ and $\mathbf{M}_t$ are defined as $\mathbf{A}_t = \nabla\mathbf{f}_t(\boldsymbol{\mu}_{t-1})$ and $\mathbf{M}_t = \nabla\mathbf{g}_t(\bar{\boldsymbol{\mu}}_t)$ where $\boldsymbol{\mu}_{t-1} = \boldsymbol{\Omega}_{t-1}^{-1}\boldsymbol{\xi}_{t-1}$. The prediction and updating steps 3 and 6 of the IF algorithm are substituted by the following expressions (Thrun et al., 2005):

$$\bar{\boldsymbol{\xi}}_t = \bar{\boldsymbol{\Omega}}_t\mathbf{f}_t(\boldsymbol{\mu}_{t-1}) \tag{3.51}$$

$$\boldsymbol{\xi}_t = \bar{\boldsymbol{\xi}}_t + \mathbf{M}_t^T\mathbf{S}_t^{-1}[\mathbf{z}_t - \mathbf{g}_t(\bar{\boldsymbol{\mu}}_t) + \mathbf{M}_t\bar{\boldsymbol{\mu}}_t] \tag{3.52}$$

This linearization can lead to underestimations of the covariance matrix for the case of highly non-linear functions.

3.3.2 The Unscented Transform

The Unscented Transform (UT) (Julier and Uhlmann, 1997a) is another approximation for the use of the Gaussian filters with non-linear transition and measurements functions. The important step in the Kalman or Information filter is the estimation of how the mean and covariances are transformed after the prediction and updating steps. That is, given a random variable $\mathbf{X}$, with mean $\bar{\mathbf{x}}$ and covariance $\boldsymbol{\Sigma}_{\mathbf{xx}}$ the problem is to estimate how the uncertainties on $\mathbf{X}$ are transferred into uncertainties over another variable $\mathbf{Y}$ that is related to $\mathbf{X}$ through the non-linear function $\mathbf{f}$, that is, how to estimate $\bar{\mathbf{y}}$ and the covariance $\boldsymbol{\Sigma}_{\mathbf{yy}}$

The UT (see Algorithm 3.5) tries to estimate these quantities drawing samples from the initial distribution on $\mathbf{X}$, transforming these samples under $\mathbf{f}$ and computing the resultant parameters of the transformed samples. The idea is quite similar to Monte Carlo-like procedures. However, in this case, the samples, called *sigma points*, are drawn in a deterministic way, indicated by lines 3 and 5 of Algorithm 3.5. The way the samples are selected ensures that the estimated first and second moments from the resultant distribution are accurate (although the actual resultant distribution

Algorithm 3.5 $(\mu_{\mathbf{y}}, \mathbf{\Sigma_{yy}}) \leftarrow$Unscented Transform$(\mu_{\mathbf{x}}, \mathbf{\Sigma_{xx}}, \mathbf{f})$

1: $X_0 = \mu_{\mathbf{x}}, W_0 = \frac{\kappa}{\kappa+n}$
2: **for** $i = 1$ to n **do**
3: $X_i = \mu_{\mathbf{x}} + (\sqrt{(\kappa+n)\mathbf{\Sigma_{xx}}})_i$/* *i-th column of the square root matrix* */
4: $W_i = \frac{1}{2(\kappa+n)}$
5: $X_{n+i} = \mu_{\mathbf{x}} - (\sqrt{(\kappa+n)\mathbf{\Sigma_{xx}}})_{n+i}$
6: $W_{n+i} = \frac{1}{2(\kappa+n)}$
7: **end for**
8: **for** $i = 0$ to $2n$ **do**
9: $Y_i = \mathbf{f}(X_i)$
10: **end for**
11: $\mu_{\mathbf{y}} = \sum_{i=0}^{2n} W_i Y_i$
12: $\mathbf{\Sigma_{yy}} = \sum_{i=0}^{2n} W_i (Y_i - \mu_{\mathbf{y}})(Y_i - \mu_{\mathbf{y}})^T$

Figure 3.3: Two heterogeneous UAVs in a fire detection and localization mission. The important information is the location of the alarms. The UAVs should avoid generating false alarms.

could be multi-modal and, therefore, the first and second moments could not represent adequately that distribution).

The UT can then be used for non-linear motion and measurement models as a kind of statistical linearization. It can be seen that the estimated mean and covariances are at least as accurate as those obtained by means of a first order Taylor expansion (Julier and Uhlmann, 1997a; Schmitt et al., 2002). Besides, the UT allows considering a more general class of functions than the usual first order expansion.

3.4 Vision-based object detection, localization and tracking with multiple UAVs.

In order to test the decentralized perception scheme presented above, the application to the detection, localization and tracking of events of interest by a fleet of UAVs is considered. This application will be also considered in other chapters of the thesis, and thus the general problem is first defined in this section.

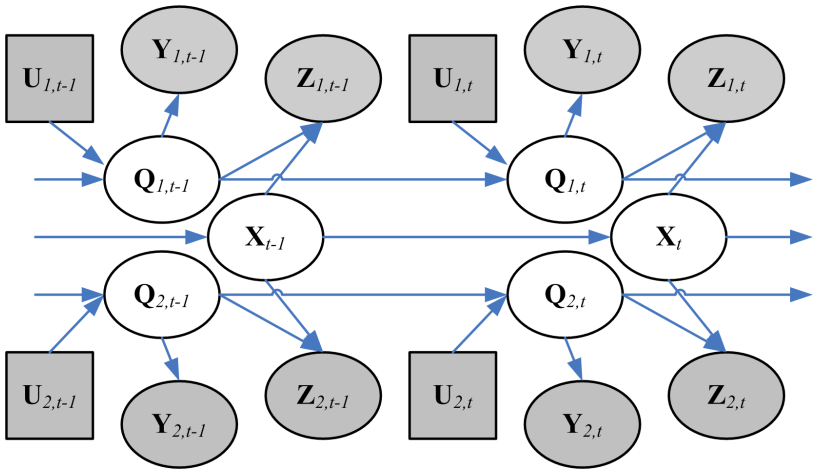

Figure 3.4: Dynamic Bayesian Network for the case of 2 UAVs. It is assumed that the data employed for UAV localization, $\mathbf{Y}$ is independent of the state of the alarms $\mathbf{X}$. However, the pose of the UAVs $\mathbf{Q}$ will influence the measurements gathered $\mathbf{Z}$.

3.4.1 Initial assumptions

Consider that a fleet of UAVs has been commanded to survey a certain area searching for a class of targets, for example fire alarms. The objective of the fleet is to detect potential targets, estimate their positions and also discard false alarms. This is one of the main objectives of the COMETS project (Ollero et al., 2005) (see Fig. 3.3).

The important information for this scenario is the location of the UAVs and the location (and perhaps other information) of the events of interest. Then, the state to be estimated in this case, in its most simple version, is comprised by the information associated to a set of N events and by the pose of the M UAVs.

From now on, it is assumed that the UAVs carry on board Global Positioning System (GPS) receivers, Inertial Measurement Units (IMUs) and other sensors for navigation. These sensors are used to localize the UAVs in a common global frame. Then, the first assumption made in this chapter is the following: the state information is factorized into two parts: the pose of the different UAVs (denoted by $\mathbf{Q}_t$) and the state of the events (denoted by $\mathbf{X}_t$). Also, the data gathered by each robot will be divided in local localization data (denoted by $\mathbf{Y}_t$) and exteroceptive data (denoted by $\mathbf{Z}_t$). Therefore, the problem of localization of the UAVs and the problem of detection and localization of events will be solved separately. The estimation of $\mathbf{Q}_t$ will be made using the data set $\mathbf{y}^t$, while the object detection and tracking will be based on the data gathered by the exteroceptive sensors (cameras in this case), $\mathbf{z}^t$. However, as is illustrated by Fig. 3.4, the conditional dependence of the measurements obtained by the cameras on the localization estimates of the robot must be taken into account. Of course, vision sensors could be used for localization. Chapter 7 will present techniques for vision-based UAV navigation.

The second assumption is that the actions of the robots will not affect the state of the objects

to be tracked, and then:

$$p(\mathbf{X}_t|\mathbf{X}_{t-1}, \mathbf{u}_t) = p(\mathbf{X}_t|\mathbf{X}_{t-1}) \tag{3.53}$$

This assumption usually holds when the objects to be tracked are not aware of the robots patrolling, for instance, if the objects of interest are fire alarms, or are not *evading* targets, but this would not hold for evading targets or opponent robots (Vidal et al., 2002b). Besides, it is assumed that the state of the alarms does not affect the readings $\mathbf{y}^t$ of the internal sensors.

3.4.2 Event state definition

In the general case of event detection and localization, the state $\mathbf{X}$ to be tracked obviously includes the position of the object $\mathbf{P}_t$. If a moving object is considered, it is convenient to include the velocity $\dot{\mathbf{P}}_t$ into the state to be estimated. This will be called the kinematic part of the state.

Further information will be also needed. An important objective in some missions is to confirm that an object belongs to a certain class within a set $\mathbf{\Gamma}$ (for instance, in the case of fire alarms detection, this set will include as classes fire alarms and false alarms). Therefore, the state will include information regarding the classification of the object. Also, in certain applications, some appearance information could be needed to characterize an event, which also can help in the task of data association between different UAVs with different cameras. This kind of information usually will be static, and will be represented by $\boldsymbol{\theta}$.

The complete state to be estimated is composed by the status of all the events, N_t, and the number of events can vary with the time. The state at time t is then represented by vector $\mathbf{X}_t = [\mathbf{X}_{1,t}^T, \ldots, \mathbf{X}_{N_t,t}^T]^T$. Each potential alarm k is defined by:

$$\mathbf{X}_{k,t} = \begin{pmatrix} \mathbf{P}_{k,t} \\ \dot{\mathbf{P}}_{k,t} \\ \boldsymbol{\theta}_k \end{pmatrix} \tag{3.54}$$

The information about the events will be inferred from all the measurements $\mathbf{z}_t$ gathered by the fleet of robots $\{\mathbf{z}_{j,t}, j = 1, \ldots, M_t\}$. One of the main issues is to determine the likelihood functions for the different sensors that can be onboard the different robots. The measurements obtained by the sensors depend not only on the environment, but also on the pose of the sensors (indeed, in Fig. 3.4 this conditional dependence is explicitly depicted). Therefore, the uncertainty in $\mathbf{Q}_t$ must be taken into account for a correct estimation of $\mathbf{X}_t$. Thus, in order to compute the likelihood function, this dependence should be marginalized out:

$$p(\mathbf{z}_{j,t}|\mathbf{X}_t) = \int p(\mathbf{z}_{j,t}|\mathbf{X}_t, \mathbf{Q}_{j,t}) p(\mathbf{Q}_{j,t}) d\mathbf{Q}_{j,t} \tag{3.55}$$

3.5 The likelihood function for vision

In this and the following chapters, the application of vision sensors for detection, localization and monitoring will be considered. UAVs will use the on-board cameras to detect events of interest and

Figure 3.5: An infrared and a visual image of a fire alarm. The objective is to fuse all the data from the different vehicles of the fleet to obtain a global belief on the parameters of a given alarm.

to track these events (see Fig. 3.5).

The key point in the Bayesian framework adopted is to determine the likelihood function $p(\mathbf{z}_t|\mathbf{X}_t, \mathbf{Q}_t)$ for the case of the cameras. In the limit, $p(\mathbf{z}_t|\mathbf{X}_t, \mathbf{Q}_t)$ would be a probabilistic model of image formation, that is, the probability that the pixels of the image have certain illuminance or colour values given the status of the environment and the position of the sensor.

In most cases, such a complex model is not needed. Instead of the raw pixels, some features related to the application are extracted from the images. Nevertheless, it is required to formulate all these steps probabilistically in order to incorporate the measurements in the estimation process.

3.5.1 Feature extraction

In the application considered (object detection and localization), the images captured by the cameras on board the UAVs should be analyzed looking for the objects of interest. Detection consists on the declaration of an event of a given class. For instance, the presence of fire, moving objects, etc.

The image processing functions should be able to segment the objects of interest on the image plane, and differentiate them from the background. Moreover, the algorithms will obtain a set of features related to the identity of the object $\boldsymbol{\theta}$. In order to include this into the probabilistic framework, it is needed to relate the visual features to the state through:

$$p(\mathbf{z}_t|\mathbf{X}_t) = p(\mathbf{z}_t|\mathbf{P}, \dot{\mathbf{P}}, \boldsymbol{\theta}), \boldsymbol{\theta} \in \boldsymbol{\Gamma} \tag{3.56}$$

These features will depend on the application considered. In general, determining the likelihood function will consists on a learning phase over the sensorial space; these learning phase should provide a mapping between features and classes.

Fire events detection

In the particular application of fire detection, the space state for the identity part of the state $\boldsymbol{\theta}$ consists only on two potential values, $\boldsymbol{\Gamma} = \{fire,\ no\ fire\}$.

Appendix B presents the algorithms used for fire segmentation. These algorithms provide directly a binary decision over the image plane, so that the measurements $\mathbf{z}_t$ are a set of blobs over the image

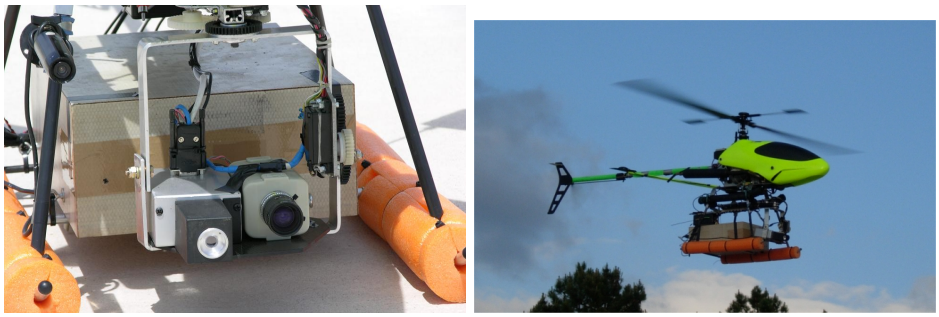

Figure 3.6: UAV carrying an electronic box that incorporate two cameras as exteroceptive sensors.

plane classified of fire. As it will be seen, it is important also to employ the negative information encoded in the regions of the image not classified as fire.

The algorithms presented in Appendix B are based on a training phase, in which classification probabilities are learnt from example images. After that, they are used for unsupervised classification of image objects. There is always the chance for *false positives* and *misdetections*. False positives occur when the algorithm detects objects but there are no objects of the considered class in the field of the camera. Misdetections happen if no response is given when an object is present. These two facts have to be used to determine the likelihood function associated to the processed images.

A simple model will be used. It consists of characterizing the segmentation algorithms by two values

- The probability of detection (P_D), defined as the likelihood that, being an object of the given class within the field of view, the object is effectively detected $p(\mathbf{z}_t = 1|\boldsymbol{\theta} = fire)$.
- The probability of false positive (P_F), defined as the likelihood that the algorithm generates a response when no actual object is within the field of view of the camera $p(\mathbf{z}_t = 1|\boldsymbol{\theta} = no\,fire)$.

3.5.2 Projective geometry

In order to complete the main aspects of the likelihood function, it is needed to relate objects on the image plane with the position of objects in the 3D world. Cameras project points in the space into points on the image plane. Cameras are usually modeled using the tools of projective geometry (Hartley and Zisserman, 2004; Faugeras and Luong, 2001). The projection is modeled by the *pin-hole* projection model. Following this model, each point in the space, $\mathbf{p}$ and its corresponding image pixel $\mathbf{m}$ on the image plane of camera are related by equation (3.57), where $\mathbf{p}$ and $\mathbf{m}$ are in homogeneous coordinates:

$$s\mathbf{m}_t = \mathbf{A}\begin{pmatrix}\mathbf{R}_t & -\mathbf{t}_t\end{pmatrix}\mathbf{p}_t \tag{3.57}$$

where $\mathbf{A}$ is the upper triangular internal calibration matrix of the camera. $\mathbf{R}_t$ and $\mathbf{t}_t$ are the rotation and translation that refer the camera coordinate system and a global reference frame, and are part

of the estimated state $\mathbf{Q}_t$ of the UAV. Several methods can be used for camera calibration (Tsai, 1987). Here, the method presented by (Zhang, 2000) is used.

Equation (3.57) implies a non-linear relation between the state and the measurements. Moreover, if the pose is uncertain, it has to be considered when obtaining the corresponding likelihood (and, also in this case, the relation among variables are non-linear). Therefore, if Gaussian filters are to be used, a linearization (either by means of Taylor expansion or by using the UT) is required.

Occlusions

One final aspect that can influence the likelihood function when vision is considered is the effect of occlusions. The 3D structure of the world should be also taken into account when relating information in the 3D world and the images.

In the applications presented in the thesis, it will be considered that the occurrence of occlusions is low, as the UAV will operate in the open field for fire detection and localization. However, a complete likelihood model should consider the potential effect of occlusions.

Putting together the geometric model of the camera with the probabilistic characterization of the segmentation algorithms allows to establish a likelihood function for the image-base measurements obtained by the UAVs. $p(\mathbf{z}_t|\mathbf{X}_t, \mathbf{Q}_t)$ would be the likelihood of getting the features extracted (that is, the objects segmented on the image plane) given the state of the events $\mathbf{X}_t$ and the position and orientation of the camera $\mathbf{Q}_t$.

3.6 Determining prior position information from cameras

Sometimes it is needed to determine the position of an object of interest from its position on the image plane. For instance, when an object is detected the first time, one should obtain an initial estimation of its position $\mathbf{P}$.

Unfortunately, relation (3.57) is not invertible. Therefore, although $\mathbf{A}$, $\mathbf{R}$ and $\mathbf{t}$ may be known, from the position on the image plane of an object $\mathbf{m}$, it is not possible to recover its 3D position $\mathbf{p}$. If nothing more is known, cameras only provide *bearing* information, and the full 3D position is not observable.

Several images from different points of view of the same object can be used to estimate the distance of a point, and thus its 3D position, by triangulation, as in stereo vision (Ayache and Sander, 1991; Faugeras, 1993). Also, if a Digital Elevation Map (DEM) of the scene is available, and the pose of the camera is known in the same coordinate frame as the DEM, it is possible to estimate the position of an object on the ground by ray tracing techniques, provided that the camera is calibrated ($\mathbf{A}$ is known). Through this chapter, it is assumed that such a map exists. In some applications, this kind of map could be provided by a UAV commanded to generate a DEM of the scenario (Hygounenc et al., 2004).

The procedure of determining the geographical position of an object of interest is also called *geolocation*. If a geolocation procedure can be applied, the UAVs provide as measurements direct estimations on the position of potential alarms.

3.7 Decentralized Information Filter for object detection and localization

After the main aspects of the problem have been stated, the chapter ends presenting a first approach for the solution of the problem of distributed object detection and localization using several UAVs.

Recalling what was presented in section 3.4, the objective is to estimate the state of the potential alarms present in an scenario. This first solution employs the Information Filter described in section 3.1.2 to estimate the position $\mathbf{P}_{k,t}$ and velocity $\dot{\mathbf{P}}_{k,t}$ of the alarms. Therefore, each UAV will maintain a multivariate Gaussian distribution (in information form) for the position and velocity of the alarms, and will update its knowledge locally from the information provided by the cameras employing Algorithm 3.4. These local estimations will be shared with the other UAVs of the fleet, which will combine them by using the relations given by (3.45) and (3.46) in order to avoid overconfident estimates in the case of a fully decentralized filter.

3.7.1 Local filters

The information maintained by each UAV i is represented by the information vector $\boldsymbol{\xi}^i$ and matrix $\boldsymbol{\Omega}^i$. In order to apply the filter, the motion and measurement models should be defined.

Motion model

In general, the motion of the alarms will not depend on the others. That is, at time t, $\mathbf{A}_t = diag\{\mathbf{A}_{1,t}, \ldots, \mathbf{A}_{N_t,t}\}$. Also $\mathbf{R}_t = diag\{\mathbf{R}_{1,t}, \ldots, \mathbf{R}_{N_t,t}\}$.

As a generic motion model of the targets, a discrete version of the continuous white noise acceleration model or second-order kinematic model is used (Bar-Shalom et al., 2001; Stone et al., 1999). In this model, the velocity is assumed to be affected by an acceleration modeled as a white noise of zero mean and with power spectral density $\mathbf{q}$. The discretized version of this linear motion model for an object k, is characterized by:

$$\mathbf{A}_{k,t} = \begin{pmatrix} \mathbf{I} & \Delta t\mathbf{I} \\ 0 & \mathbf{I} \end{pmatrix} \tag{3.58}$$

and

$$\mathbf{R}_{k,t} = \begin{pmatrix} \frac{1}{3}\Delta t^3\mathbf{I} & \frac{1}{2}\Delta t^2\mathbf{I} \\ \frac{1}{2}\Delta t^2\mathbf{I} & \Delta t\mathbf{I} \end{pmatrix} \mathbf{q} \tag{3.59}$$

for Δt the time difference between consecutive steps.

Measurement model and likelihood function

The segmentation algorithms will provide the position of potential alarms on the image plane. If the alarms are located on the ground, and if the geolocation procedure mentioned can be applied,

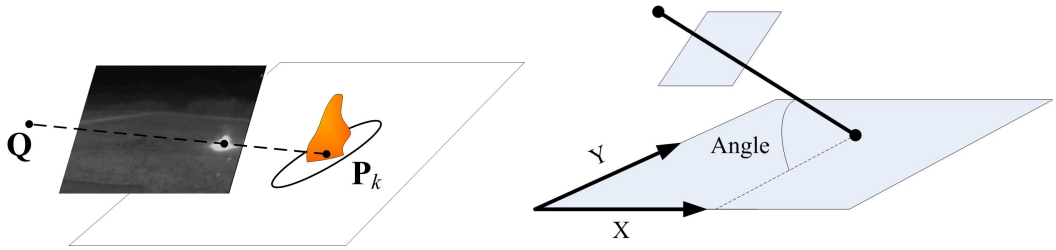

Figure 3.7: Left: the uncertainties on robot pose will translate into uncertainties on the estimated alarm position. Right: scheme of the error analysis.

each UAV can determine directly the 3D position of the objects segmented on the image plane. If the current measurement $\mathbf{z}_t$ is associated to a particular alarm k, then:

$$\mathbf{z}_t = \mathbf{M}_{k,t} \begin{pmatrix} \mathbf{P}_{k,t} \\ \dot{\mathbf{P}}_{k,t} \end{pmatrix} + \varepsilon_t, \tag{3.60}$$

and

$$\mathbf{M}_{k,t} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \end{pmatrix} \tag{3.61}$$

The errors $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{S}_t)$ in the estimated position arise due to the errors on the position and orientation of the sensor ($\mathbf{Q}_t$) and the terrain model (see Fig. 3.7, left). Moreover, the geolocation procedure is non-linear. In order to propagate the uncertainties in $\mathbf{Q}_t$ and obtain an estimation of the covariances of the error in (3.5) the Unscented Transform is used.

Other options could be used, as for instance a first-order linealization (see Section 3.3). As stated there, the Unscented Transform is more accurate than a first order expansion. As a comparison, some simulation results are presented. The mean and covariance of the errors for an object located in the centre of the image are estimated, for the case of a planar scene. The estimations are obtained by first order Taylor linearization of the backprojection equations over the plane (derived from (3.57)), using the Unscented Transform and also by a Monte Carlo simulation. The camera distance is kept to 40 m. to the plane, and the angle of view is varied (see Fig. 3.7 right)[3].

Figure 3.8 shows the estimated variances on the errors using a first order expansion (as in the EKF) and the Unscented Transform, and compared with a Monte Carlo estimation for the position of an object detected on the image plane, and a planar surface.

Figure 3.9 shows the Kullback-Leibler divergence (Kullback and Leibler, 1951) (see Appendix A) between the MC estimated distribution and the first-order and UT based estimations.

Therefore, the UT is used to determine the mean $\mathbf{z}_t$ and covariance matrix $\mathbf{S}_t$ on the estimated position for any alarm detected on the image plane.

[3]the calibration matrix of the camera is $\mathbf{A} = [347.881, 0, 119.512; 0, 345.876, 161.781; 0, 0, 1]$, corresponding to an actual infrared camera employed in the experiments

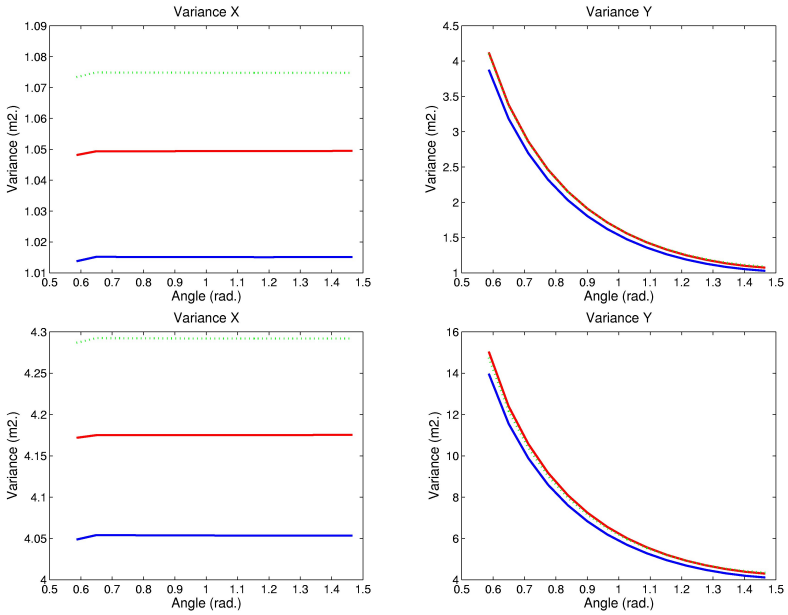

Figure 3.8: Variances in the estimated X and Y coordinates. Red: Monte Carlo estimation. Solid: First-order Taylor expansion. Dotted: UT.

Local information filter with perfect data association

The previous equations are then used locally by each UAV to update their local beliefs about the current alarms. A further assumption allows to simplify the algorithm. As the motion model and motion noise covariances are block diagonal, if the information matrix is block-diagonal $\mathbf{\Omega}^i_{t-1} = diag\{\mathbf{\Omega}^i_{1,t-1}, \ldots, \mathbf{\Omega}^i_{N_t,t-1}\}$ each alarm k can be predicted separately, leading to a (local) parallelized computation.

Using its cameras, an UAV can obtain more than one measurement each time (see Fig. 3.10). If the set of measurements can be associated with the current set of alarms, so that any measurement is associated to an alarm k, and if the measurements are independent (and thus, $\mathbf{S}_t$ is block diagonal), then the information matrix is block diagonal at any given time t, $\mathbf{\Omega}^i_t = diag\{\mathbf{\Omega}^i_{1,t}, \ldots, \mathbf{\Omega}^i_{N_t,t}\}$, and therefore each alarm can be updated independently: that is, the global Information Filter for all the alarms can be divided into N Information Filters, a separated one for each alarm.

However, the position measurements about the different alarms obtained by one UAV are not independent. When obtaining the position of the alarms, the errors on the UAV position $\mathbf{Q}_t$ induce errors on the estimated position of the alarms, errors that are correlated for all the alarms detected, as is illustrated in Fig. 3.10. Therefore, the dependencies among measurements should be explicitly marginalized out if one wants to keep the alarms updated independently. The marginalization is straightforward for the case of Gaussian measurements (see Appendix A). It should be noted that getting rid of measurements dependencies reduces the amount of storage required and the computational burden, but it comes at a cost. Some information is lost, the relation between alarms.

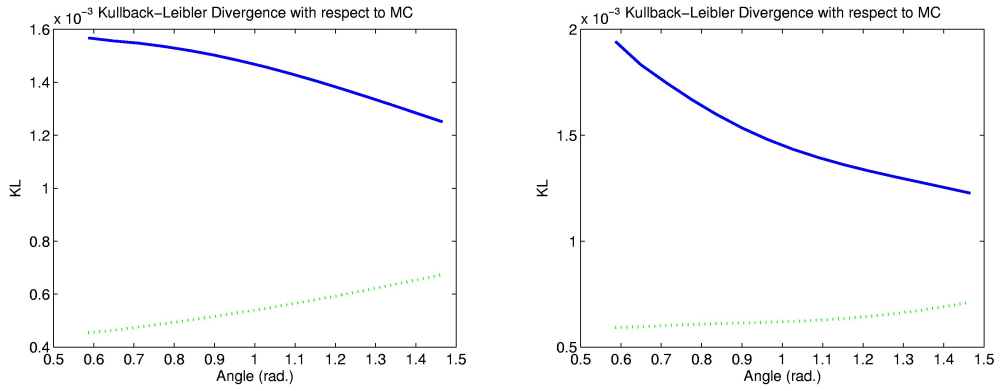

Figure 3.9: Kullback-Leibler divergence with respect to the MC estimation. Solid: First-order expansion. Dotted: UT estimation.

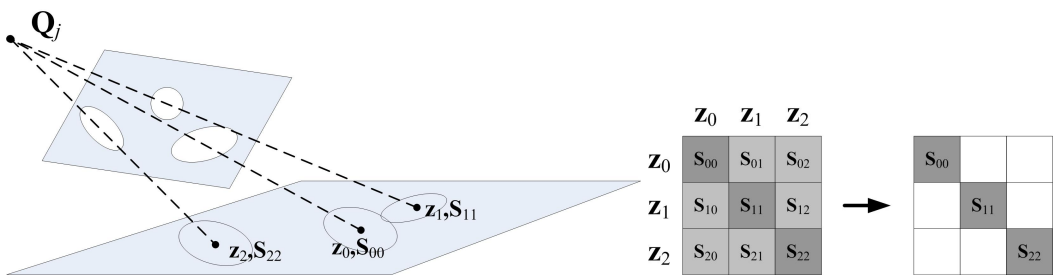

Figure 3.10: Several alarms can be detected on the image plane. The errors in $\mathbf{Q}_t$ induce correlated errors in the estimated position of the alarms $\mathbf{z}_t$ (left). If the alarms are to be evolved independently, the dependencies should be marginalized out (right).

This information would allow to propagate information about one alarm to the other related (as in the SLAM problem).

Prior belief

The only issue to be described is how alarms are initialized. When an UAV segments an alarm, the geolocation procedure directly provides the initial values for $\boldsymbol{\xi}_k$ and $\boldsymbol{\Omega}_k$.

$$\boldsymbol{\Omega}_k = \begin{pmatrix} \mathbf{S}_t^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \tag{3.62}$$

and

$$\boldsymbol{\xi}_k = \begin{pmatrix} \mathbf{S}_t^{-1}\mathbf{z}_t \\ \mathbf{0} \end{pmatrix} \tag{3.63}$$

Final decentralized filter

The final decentralized filter is given by Algorithm 3.4 and the previous definition of the prediction and updating functions. One filter is considered for each alarm k. The information that each UAV i maintains over this alarm is given by $\boldsymbol{\xi}^i_{k,t}$ and $\boldsymbol{\Omega}^i_{k,t}$ ($\boldsymbol{\xi}^{t,i}_k$ and $\boldsymbol{\Omega}^{t,i}_t$ in the dynamic case).

In the dynamic case, special care should be taken when receiving information from other UAV. The trajectories should be adjusted so that the state space is the same in both cases by using equations (3.33).

3.8 Data association

The previous sections have presented the filter under the assumption of perfect data association. That is, each measurement is associated to one particular alarm k, or when an UAV receives information about an alarm from other UAV, it knows to which local alarm it belongs.

The data association problem tries to determine what measurements correspond to what alarms, or what alarm corresponds to what alarm when combining beliefs received from other UAVs. The first one is usually called scan-to-track association, while the later is called track-to-track association. This general problem is known to be NP-hard in the general case. As the main objective is to show cooperative characteristics, the approach followed is fairly simple, although more complex ones could be used.

3.8.1 Scan-to-track association

The first approach is to use a gated nearest neighbor technique (Feder et al., 1999). The measured object $\mathbf{z}_{p,t}$ is associated with the local alarm k if they satisfy:

$$d^2_{kp,t} = [\mathbf{M}_{k,t}\boldsymbol{\mu}_{k,t} - \mathbf{z}_{p,t}]^T \boldsymbol{\Sigma}^{-1}_{kp,t}[\mathbf{M}_{k,t}\boldsymbol{\mu}_{k,t} - \mathbf{z}_{p,t}] \leq d^2_{th} \tag{3.64}$$

where $\boldsymbol{\Sigma}_{kp,t} = \mathbf{M}_{k,t}\boldsymbol{\Omega}^{-1}_{k,t}\mathbf{M}^T_{k,t} + \mathbf{S}_{p,t}$. If $\mu_{k,t}$ and $\mathbf{z}_{p,t}$ are Gaussian, then $d^2_{kp,t}$ follows a χ^2 distribution. d^2_{th} is chosen as 9 (less than 0.1 probability that a correct association gives a greater value (Feder et al., 1999; Bar-Shalom and Fortmann, 1988)). If there are more than one measure provided by the same UAV that accomplish (3.64), then the nearest (that of minimum $d^2_{kp,t}$) is chosen as the correct association with alarm k.

3.8.2 Track-to-track association

When an UAV receives the belief from another one, it also should determine to which local alarm it belongs. Each alarm is marked with a label, but the labels are not synchronized, so the UAV must determine the conversion.

This is also accomplished by using a nearest neighbor technique.

$$d^{ij^2}_{kp,t} = [\boldsymbol{\mu}^i_{k,t} - \boldsymbol{\mu}^j_{p,t}]^T \boldsymbol{\Sigma}^{ij^{-1}}_{kp,t}[\boldsymbol{\mu}^i_{k,t} - \boldsymbol{\mu}^j_{p,t}] \leq d^2_{th} \tag{3.65}$$

Alarm p from UAV j and k from UAV i are associated if they minimize the previous distance, where $\mathbf{\Sigma}^{ij}_{kp,t} = \mathbf{\Sigma}^{i}_{k,t} + \mathbf{\Sigma}^{j}_{p,t}$. However, once a remote track has been assigned to a local one, a look-up table that relates the labels of the remote and local UAVs is created, and used from that time on for data association.

3.8.3 The dynamic case

Again, the previous techniques assume that the measurements obtained from the cameras or the belief received from other UAVs are correctly matched to the corresponding alarm.

Scan-to-track association

Together with the full trajectory, the filtering distribution $bel(\mathbf{X}_t) \sim \mathcal{N}(\mathbf{\Omega}_t^{-1}\boldsymbol{\xi}_t, \mathbf{\Omega}_t^{-1})$ is also maintained for each alarm. For the measurements obtained from the local sensors, the same nearest neighbor technique explained before is used, considering the current estimated state.

Track-to-track association

The same approach as in the static case can be used to match the estimated trajectory received from UAV j with the local trajectories. In order to compute the Mahalanobis distance, it is needed to recover the mean $\boldsymbol{\mu}^i$ for every alarm by solving the system $\mathbf{\Omega}^i\boldsymbol{\mu}^i = \boldsymbol{\xi}^i$. Also, it is needed to obtain the inverse of $\mathbf{\Sigma}^{ij}$. The information matrix $\mathbf{\Omega}^i$ for the case of the full trajectory (and, thus, matrix $\mathbf{\Sigma}^{ij}$) can be of the order of hundred of rows/columns (for instance, for a 20 second trajectory and if each block row corresponds to one second, the matrix is 120×120).

However, the information matrix is very structured, which allows for efficient algorithms for matrix inversion. In (Asif and Moura, 2005), the authors show how, for a symmetric tridiagonal block matrix, there are algorithms nearly two orders of magnitude more efficient than direct inversion.

Again, in the case of track-to-track association, once the same identifier has been associated to a local alarm more than a certain number of times, the track is definitely associated, so that the data association problem becomes straightforward. However, this comes at cost that sometimes this could lead to wrong assignments. Nevertheless, the higher dimensionality of the state space makes it more unlikely that two different associations are compatible with the distance gate.

3.9 Results

This section presents some results on the application of the decentralized information filter for multi-UAV localization and tracking. Several experiments in simulation have been carried out. The simulations are done considering the multi-robot architecture described in Chapter 8. Moreover, the section presents results on the cooperative localization of fire alarms using data obtained during the experiments also described in Chapter 8.

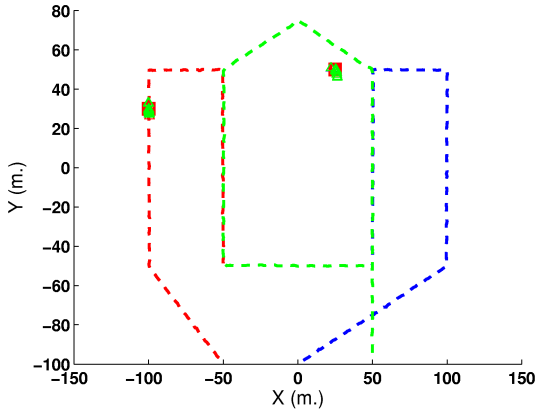

Figure 3.11: Map showing the trajectories of the vehicles and the position of the two static events for Experiment 1.

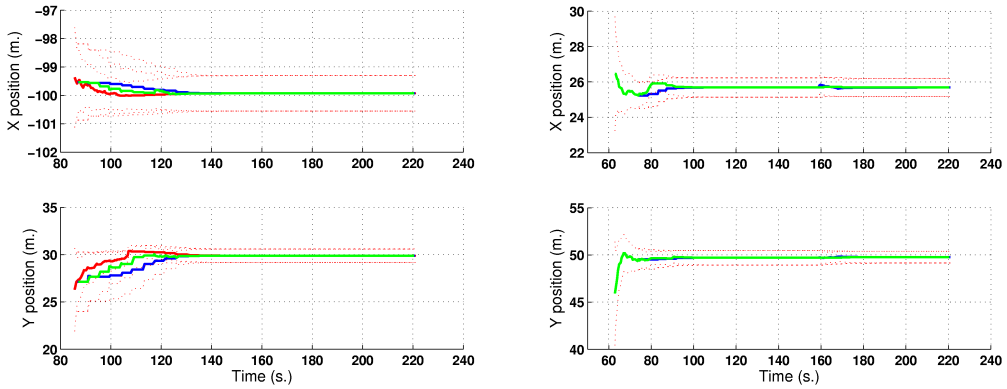

Figure 3.12: Estimated position of both static alarms for the three vehicles in Experiment 1. It can be seen how the three UAVs end with the same belief state about the alarms.

3.9.1 Experiment 1

In the first experiment, three simulated UAVs are considered, and two static alarms. One of the vehicles flies at a higher altitude, while the others are given a certain area to patrol. Figure 3.11 shows the trajectories and the position of the alarms.

The UAVs carry on board cameras and process the images, segmenting potential objects on the image plane. The positions of these objects on the ground are obtained and integrated into a local IF to estimate the position of potential alarms. The UAVs also apply the fusion algorithms described above in order to incorporate the information received from their companions. A fully decentralized approach is considered, and thus the Covariance Intersection algorithm is used to fuse the information.

Figure 3.12 shows the estimated position of the alarms for the three UAVs. It can be seen how

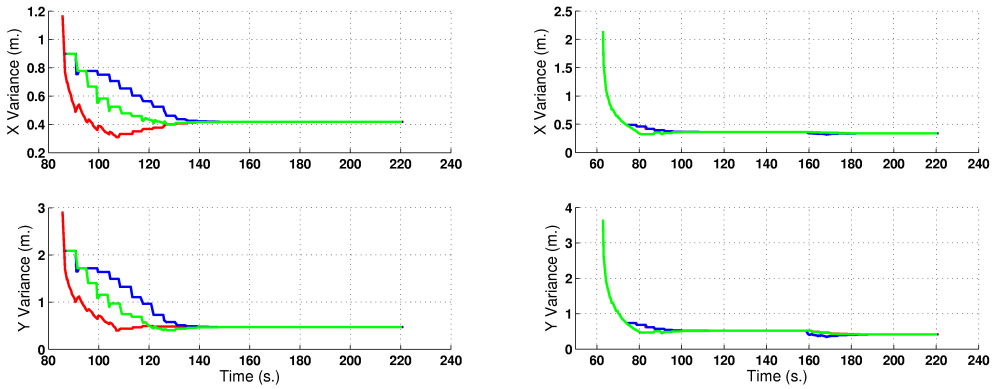

Figure 3.13: Evolution in the estimated standard deviation of the errors for the three vehicles in Experiment 1.

the three UAVs converge to the same belief, which would be the same as a centralized version but for the conservative fusion rule employed. Figure 3.13 shows the estimated standard deviation on each coordinate for the three UAVs.

3.9.2 Experiment 2

The second experiment consists of the detection and tracking of a moving object by three UAVs. The three UAVs are commanded to fly to some points and stay hovering looking to the object. Figure 3.14 shows the estimated position of a moving object by the three UAVs using cameras. As the object is moving, each UAV maintains the belief over 20 seconds of object trajectory (see Fig. 3.15).

Figure 3.16 shows a detail of the estimated position against the actual one. Also, the variances in the X and Y coordinates are shown. It can be seen how one UAV, after losing the object from its field of view, increases its uncertainty about the position of the object. This uncertainty is drastically decreased when it receives information from a second UAV that perceives the object. The third UAV is able to maintain an estimation on the position of the object even if it has not perceived it directly. At the end of the experiment, the object abandons the field of view of the UAVs and no further data is received, so the uncertainty about the position of the object increases.

Figure 3.17 shows also the estimated velocity of the object. Finally, Fig. 3.18 shows the times and sizes of the messages exchanged. In this particular setup, the UAVs communicate with their neighbors each 5 seconds, provided that they have updated their information with their local sensors.

Each point in the trajectory stores $6 \times D$ bytes for the mean position and velocity (D being the size in bytes of a double precision number) and $(21+36) \times D$ bytes for the elements of the tridiagonal block and symmetric information matrix (the values needed for storing $\mathbf{\Omega}_{tt}$ and $\mathbf{\Omega}_{tt-1}$, see Section 3.1.3). A time stamp is also stored for each point of the trajectory.

Considering that a 20 second trajectory is exchanged, with intervals of 1 second within the

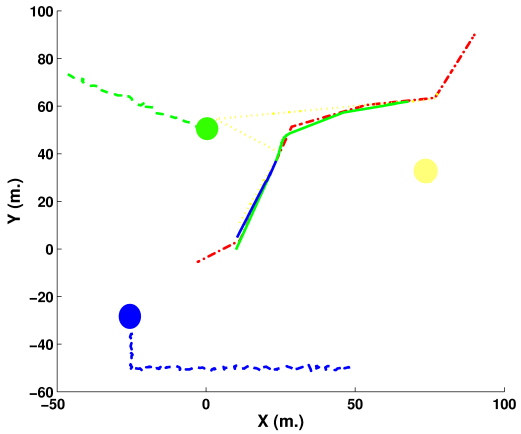

Figure 3.14: Map showing the trajectory of the event (dash-dotted) and the position of the UAVs (circles) for Experiment 2. The estimated trajectory of the object is also shown (solid).

trajectory, and for $D = 8$ bytes, the message size is around 10 kB. One of the advantages of receiving the full trajectories is that the system admits latency in the reception of the information, so therefore the bandwidth required for transmitting this information can be of the order of few kBytes per second.

3.9.3 Experiment 3

The third experiment is similar to Experiment 2. It involves two UAVs and a mobile object describing a more complex trajectory. In this case, the two UAVs are permanently within communication range. Figure 3.19 shows the trajectory of the object and the UAVs.

The estimated position and covariances are shown in Fig. 3.20. Figure 3.21 compares the estimated velocity with the actual one.

3.9.4 Experimental results on cooperative fire detection using UAVs

The last experiment described corresponds to an actual flight of two UAVs in a fire detection mission. In the experiment, a fire was lit at a certain position, recorded by using a hand-held GPS receiver.

Firstly, an UAV is sent to cover the area. Once the UAV detects a potential alarm, a second one is sent over the same zone. Figure 3.22 shows the trajectories of the UAVs and the final estimated position of the alarm.

The second UAV integrates the information received from the first one with the local measurements obtained with an infrared camera. Figure 3.23 shows the evolution on the estimated position for this second UAV, compared with the actual position of the alarm. It can be seen how the estimated position is consistent and quite accurate. The experiments and the configuration of the UAVs are more thoroughly described in Chapter 8.

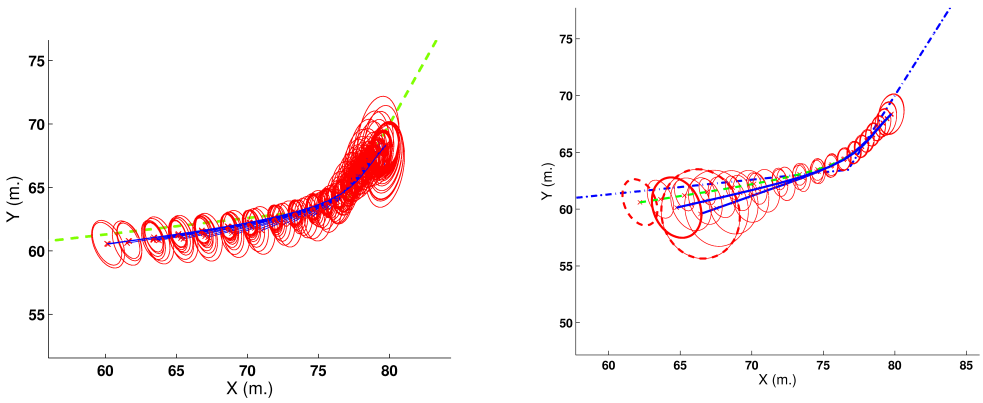

Figure 3.15: Experiment 2. In the dynamic case, the UAVs maintain a belief over the state trajectory. Left: using local information, each UAV updates the full trajectory (only information on the position is shown, but the velocity is also considered). Right: when a trajectory is received from other UAV (dashed), it is fused with the local one. The result is given by the solid line.

Figure 3.24 shows one of the infrared images of the second UAV. It can be seen how a second false alarm is also detected. In this chapter, the status of the alarm $\boldsymbol{\theta}$ has not been considered. Next chapter will show how this information can be included. By doing this, it is also possible to discard potential fire alarms.

3.10 Discussion

This chapter has described how Gaussian filters can be decentralized. In this case, there are analytical tools that allow to avoid overconfident estimations due to rumor propagation. The use of Gaussian filters for decentralized estimation is not new and has been presented before (Sukkarieh et al., 2003; Nettleton et al., 2003). However, the exploitation of the sparse structure of the information filter for cooperative multi-UAV cooperative tracking of mobile objects is considered novel.

Also, this chapter has presented the particular problem of multi-UAV detection and localization of events. The described filter has been applied to determine the position and velocity of the objects of interest. Simulation results are presented and also results obtained with actual UAVs in fire detection missions. These results will be complemented by the results on multi-UAV fire detection and localization that will be presented in Chapter 8.

The Gaussian filters can be only applied in certain situations. But they cannot be used for instance to estimate discrete states, as for instance if the object belongs to a certain class from the views of several vehicles, that is, to estimate $\boldsymbol{\theta}$ from the measurements. Also, they cannot incorporate negative information. Thus, the next chapter considers a different approach to decentralized Bayesian estimation. Also, it will be seen how both approaches can be mixed in order to estimate different information about the same event.

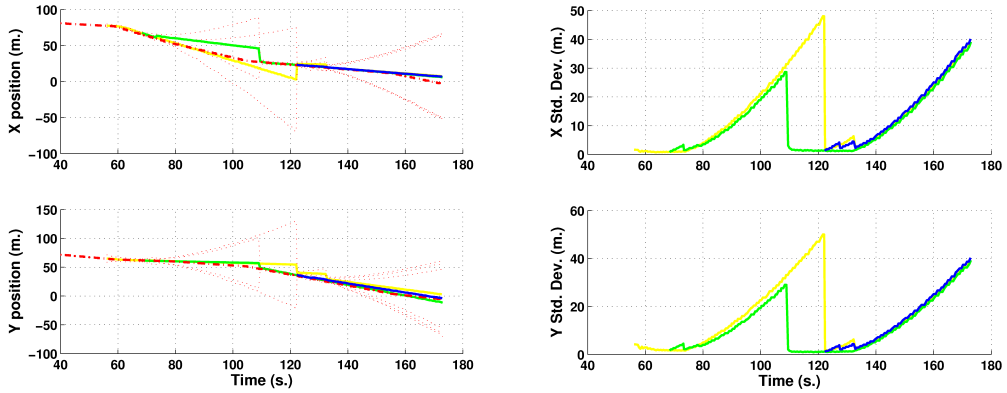

Figure 3.16: Estimated position (X and Y axis), and estimated variances for both values for the three UAVs of Experiment 2.

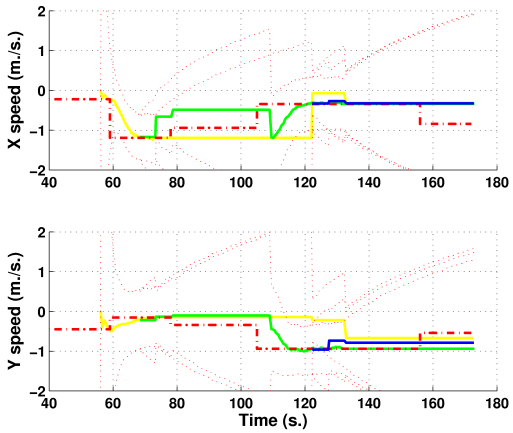

Figure 3.17: Estimated velocity by the three UAVs compared to the actual one (dash-dotted).

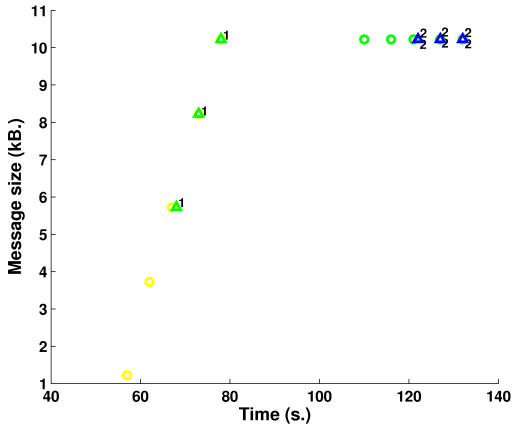

Figure 3.18: Messages exchanged between the UAVs in Experiment 2. The circles indicate transmission, while the triangles reception (and the number, the UAV from which the information is received). Each UAV transmits each 5 seconds if it has new information (that is, if it has performed at least one update on the belief due to local information).

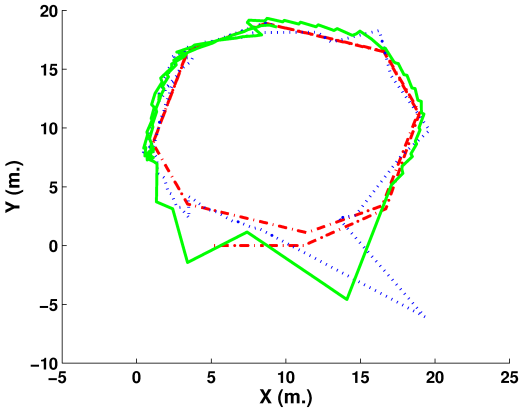

Figure 3.19: Map showing the actual trajectory of the object (dash-dotted) and the estimated trajectory by the UAVs.

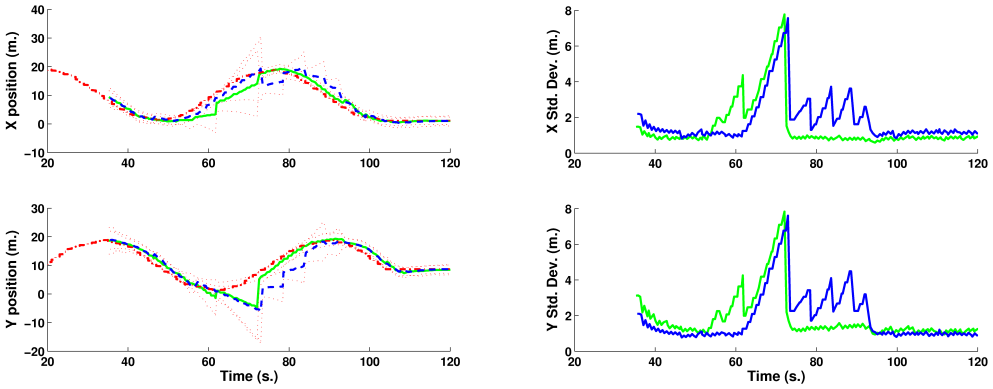

Figure 3.20: Left: Estimated position of the object (solid) compared to the actual one (dashed). Right: evolution of the standard deviation on the position.

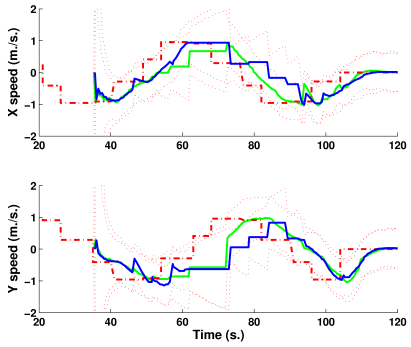

Figure 3.21: Estimated velocity by the two UAVs compared to the actual one.

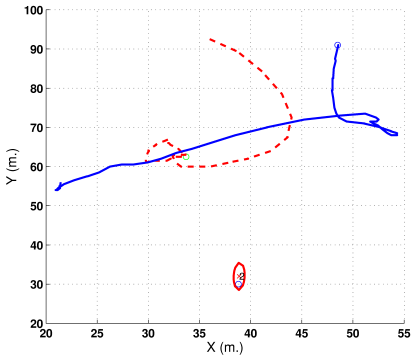

Figure 3.22: Trajectories of the UAVs. The solid represents the trajectory of the first one, which initially detects one alarm. The second UAV (dashed) receives the information from the first and fuses it with its local data. The ellipse represents the final estimated position of the alarm.

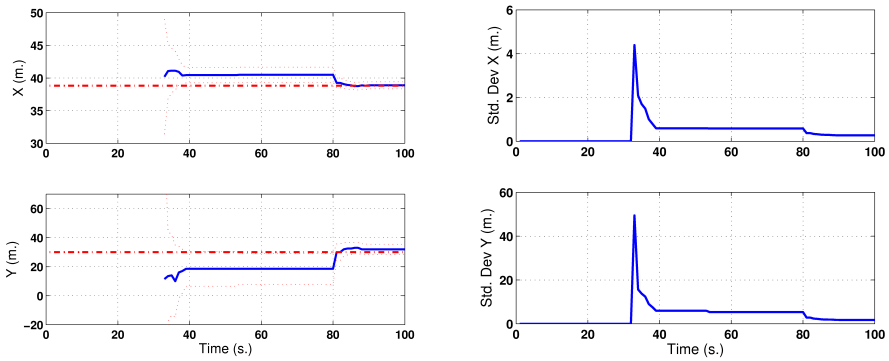

Figure 3.23: Evolution of the estimated mean position (left) and the standard deviation (right). The actual position of the alarm is shown dash-dotted. Around time 30, the first UAV detects the alarm and decreases its uncertainty on its position by using its camera. Then, the second UAV reaches the place and around time 80 integrates its own measurements.

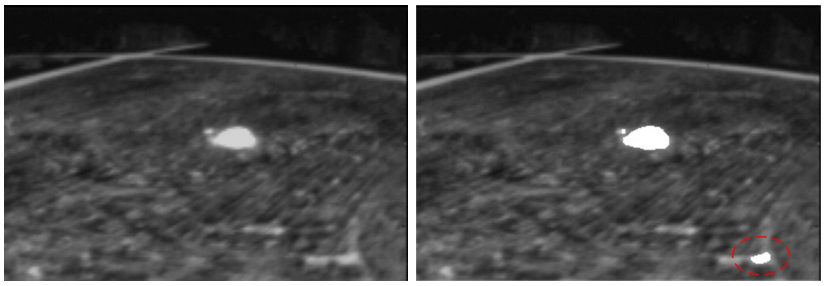

Figure 3.24: Infrared image from one UAV and the segmentation results. It can be seen how a false alarm is detected (marked with an ellipse). The results only show how the position of the actual alarm is estimated. The Gaussian filters cannot incorporate negative information in order to discard those false alarms.

There's a 68.71 chance you're right.
End of Line.
MASTER CONTROL PROGRAM
Tron (1982)

Chapter 4

Decentralized perception employing grid-based representations

The use of parametric distributions, like the multivariate Gaussian, for representing the belief state has the advantage of keeping the number of required parameters low, and therefore the storage and bandwidth needed to transmit the belief state to other robots. Another advantage of the Gaussian distributions is that there is an analytical solution for the removal of common information, and the Covariance Intersection formula gives a conservative way of combining beliefs even in the case that this common information is unknown.

However, decentralized perception using the Information Filter has some drawbacks. The Information Filter (and its dual, the Kalman Filter), requires some restrictions that in some cases do not hold. Firstly, they are restricted to Gaussian distributions for the belief state $p(\mathbf{X}_t|\mathbf{z}^t, \mathbf{u}^t)$, and thus they cannot handle multi-modal distributions, which for instance arise when several hypotheses should be maintained at the same time. Secondly, in order to maintain the Gaussian assumption, the measurements should be a linear function of the state, which is clearly not the case for vision-based measurements, due to the projection operation of (3.57). Furthermore, the previous method is suitable for sensors that provide *contacts*, that is, sensors in which the raw data can be segmented into information that can be associated to a particular alarm (this leads to the problem of data association, that has to be taken into account (Stone et al., 1999)). However, there are sensors that do not provide information that can be segmented into objects, as the fire detectors that will be presented in this Chapter. Also, the Gaussian filters cannot incorporate *negative information*, like a camera not seeing an object.

An alternative approach is to use non-parametric representations of the posterior. These representations lead to non-parametric implementations of the Bayes filter of Section 2.2.2. There are two main non-parametric representations used in robotics for state estimation: the histogram filter and the related evidence grids (Moravec, 1989; Elfes, 1989; Thrun et al., 2005) and particle filters (Gordon et al., 1993; Doucet et al., 2001).

Algorithm 4.1 $\{p_{k,t}\} \leftarrow$ Discrete_Bayes_Filter$(\{p_{k,t-1}\}, \mathbf{z}_t)$

1: **for** $k = 1$ to L **do**
2: $\quad \bar{p}_{k,t} = \sum_{j=1}^{L} p(\mathbf{X}_t = \mathbf{x}_{k,t} | \mathbf{X}_{t-1} = \mathbf{x}_{j,t-1}) p_{j,t-1}$
3: $\quad \bar{p}_{k,t} = \bar{p}_{k,t} p(\mathbf{z}_t | \mathbf{X}_t = \mathbf{x}_{k,t})$
4: **end for**
5: $p_{k,t} = \bar{p}_{k,t} / \sum_{j=1}^{L} \bar{p}_{j,t}$

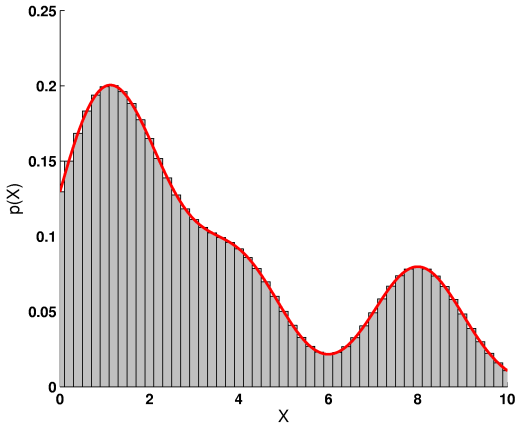

Figure 4.1: Grid-based representations can be also used to approximate beliefs over continuous states.

In this chapter, the use of evidence or certainty grids for multi-UAV perception is considered. Next chapter will deal with particle-based representations. Again, the final objective is to develop decentralized algorithms that can scale with the number of robots.

4.1 Certainty grids

4.1.1 Discrete Bayes filter

The discrete Bayes filter is employed when the state $\mathbf{X}_t$ can only take values within a finite set of discrete and disjoint events $\mathbf{X}_t \in \{\mathbf{x}_{1,t}, \ldots, \mathbf{x}_{k,t}, \ldots, \mathbf{x}_{L,t}\}$. For instance, if the state corresponds to the identity of an object from a potential set, as in $\mathbf{X}_t \in \{\text{car}, \text{person}, \text{other}\}$ or $\mathbf{X}_t \in \{\text{fire}, \text{no fire}\}$, or the state of a door $\mathbf{X}_t \in \{\text{open}, \text{close}\}$. In this case, the basic integrals of the Bayes filter (2.7) are replaced by summations. Representing by $p_{k,t} = bel(\mathbf{X}_t = \mathbf{x}_{k,t})$, the Bayes filter that updates the current belief given new data is given by Algorithm 4.1.

Although the histogram filter can only be applied to discrete state spaces, it is also employed to approximate beliefs over continuous state spaces, by partitioning this space into a set of bins and estimating the belief for each bin (see Fig. 4.1).

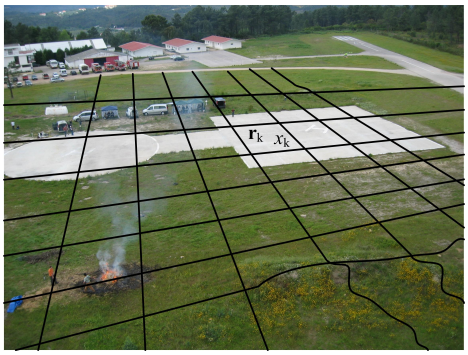

Figure 4.2: Definition of the grid for UAV object detection.

4.1.2 Binary and static certainty grids

A typical exploration mission requires that an UAV or a fleet of UAVs determine the presence or absence of certain static objects or situations within an area (for instance, obstacles, the presence of fire alarms, etc). A common approach is to divide the scenario to be explored into cells, in what is called a certainty or evidence grid (Moravec, 1989; Elfes, 1989). To each cell k, a discrete binary random variable X_k is attached, representing the presence or absence of the event considered. The variable X_k can have two values. The probability that there is an object at cell k is represented by $p(X_k = 1)$ or $p(x_k)$. Denoting by $\bar{x}_k$ the fact that there is no object at cell k, then, by definition, $p(X_k = 0) = p(\bar{x}_k) = 1 - p(x_k)$. Besides, each cell of the grid has a 3D position associated, $\mathbf{r}_k$. Figure 4.2 shows an scheme of the approach for a UAV exploration mission.

The objective of the robot team is to update the probability of the different cells. At each cell, the state is discrete and therefore Algorithm 4.1 can be employed to update the current knowledge with the new data (line 2 is not needed as the state is also static). However, as only two values are possible, the filter takes a particular simple form if the belief state is stored by using its log-odds $\log \frac{bel(x_{k,t})}{bel(\bar{x}_{k,t})}$:

$$\begin{aligned} bel(x_{k,t}) &= p(x_k|\mathbf{z}^t) = \frac{p(\mathbf{z}_t|x_k)p(x_k|\mathbf{z}^{t-1})}{p(\mathbf{z}_t|\mathbf{z}_{t-1})} \\ bel(\bar{x}_{k,t}) = 1 - bel(x_k) = p(\bar{x}_k|\mathbf{z}^t) &= \frac{p(\mathbf{z}_t|\bar{x}_k)p(\bar{x}_k|\mathbf{z}^{t-1})}{p(\mathbf{z}_t|\mathbf{z}_{t-1})} = \frac{p(\mathbf{z}_t|\bar{x}_k)[1 - p(x_k|\mathbf{z}^{t-1})]}{p(\mathbf{z}_t|\mathbf{z}_{t-1})} \end{aligned} \tag{4.1}$$

$$\frac{bel(x_{k,t})}{1 - bel(x_{k,t})} = \frac{p(\mathbf{z}_t|x_k)}{p(\mathbf{z}_t|\bar{x}_k)} \frac{bel(x_{k,t-1})}{1 - bel(x_{k,t-1})} = \prod_{\tau=0}^{t} \frac{p(\mathbf{z}_\tau|x_k)}{p(\mathbf{z}_\tau|\bar{x}_k)} \frac{bel(x_{k,0})}{1 - bel(x_{k,0})} \tag{4.2}$$

If $l_{k,t} = \log \frac{bel(x_{k,t})}{1-bel(x_{k,t})}$, then the local algorithm for updating all the cells of the grid with the new info is given by Algorithm 4.2. The probability of a given cell can be easily recovered from the log-odds:

$$bel(x_{k,t}) = 1 - (1 + e^{l_{k,t}})^{-1} \tag{4.3}$$

Algorithm 4.2 $\{l_{k,t}\} \leftarrow$ Binary_LogOdds_Filter$(\{l_{k,t-1}\}, \mathbf{z}_t)$

1: **for all** cells k **do**
2: $\quad l_{k,t} = l_{k,t-1} + \log \frac{p(\mathbf{z}_t|x_k)}{p(\mathbf{z}_t|\bar{x}_k)}$
3: **end for**

4.1.3 Decentralized filter in the multi-robot case

If all data provided by the robots $\mathbf{z}_t^m$ are received in a centralized node, under the assumption of conditionally independent data $p(\mathbf{z}_t^m|x_{k,t}) = \prod_j p(\mathbf{z}_{j,t}|x_{k,t})$, the updating equation of the filter is substituted by:

$$l_{k,t} = l_{k,t-1} + \sum_j \log \frac{p(\mathbf{z}_{j,t}|x_k)}{p(\mathbf{z}_{j,t}|\bar{x}_k)} \tag{4.4}$$

This filter is easily decentralized. If $l_{k,t}^i$ denotes the belief of UAV i for each cell k, the resultant filter is described by Algorithm 4.3. The main idea is that each robot i computes part of the running total of (4.4) considering only its local data, and sends to the other robots its own belief in log-odds form.

When a robot i receives information from another robot j it has to remove the information $l_{k,t}^{ij}$ it has in common with robot j prior to incorporating its information (line 13). This common information is due to previously received information from robot j or previous information sent to robot j. The common information is computed in lines 14 and 22 of the algorithm.

Removing unknown common information

Algorithm 4.3 will suffer from rumor propagation if the topology of the connections is not tree-shaped, as the fusion rule will not remove information received through different paths.

For instance, if UAV i uses as prior for each cell 0.5 (that is, $l_{k,t}^i = 0$), and it receives information from UAV j with a log-odds for this cell of $l_{k,t}^j = 0.847$ (corresponding approximately to a probability of 0.7), it will update its own belief to a probability of 0.7. However, if the same information is received through a second UAV (with which UAV j has communicated), then it will update its belief to $l_{k,t}^i = 2 \times 0.847$, which corresponds to a probability of 0.845 for that cell, and therefore leading to an overconfident estimation of the probability.

In the case of discrete filters, it is known that the filters will be consistent (that is, will converge to the actual distribution given enough observations) provided that positive probability mass is put on all the potential discrete events (Ghosal, 1999); that is, in the case of binary states, given that the probability $p(x_k)$ is not 1 or 0. This could happen due to numerical accuracy, and therefore it should be avoided, for instance imposing a maximum and minimum possible values for the probabilities on the cells. In this case, the filter would finally converge to the correct belief given sufficient looks at each cell.

It is not possible, however, to explore each cell more than a certain amount of time. Moreover, in exploration missions, alarms will be raised for those cells in which the estimated probability is over a given threshold. For instance, in the previous example, if an alarm is declared for any cell

Algorithm 4.3 Decentralized_Grid_Multi_Robot(i)

```
1: for all k do
2:    l^i_{k,0} = 0/* Prior */
3:    for all UAV j do
4:       l^{ij}_{k,0} = 0
5:    end for
6: end for
7: while true do
8:    for all k do
9:       if New data z_{i,t} then /* Local information */
10:         l^i_{k,t} = l^i_{k,t-1} + log p(z_{i,t}|x_k) − log p(z_{i,t}|x̄_k)
11:      end if
12:      if New belief from UAV j then
13:         Δl^i_{k,t} = l^j_{k,t} − l^{ij}_{k,t-1}/* Remove common info */
14:         l^{ij}_{k,t} = l^j_{k,t}
15:         l^i_{k,t} ← l^i_{k,t} + Δl^i_{k,t}/* Add increase of evidence */
16:      end if
17:   end for
18:   if Communication event then
19:      for all UAV j within comm. range do
20:         Send belief to UAV j
21:         for all k do
22:            l^{ij}_{k,t} = l^i_{k,t}/* Update common information */
23:         end for
24:      end for
25:   end if
26: end while
```

whose probability is over 0.8, the rumor propagation will produce raising a potential false alarm. Therefore, the rumor propagation effect should be avoided.

Other fusion rules are possible. One of them is what is called by some authors a dictatorship, in which, from the opinions received, the most informative one is chosen, discarding the rest. In the case of Bernouilli variables, for instance the opinion with less entropy or less variance could be chosen. However, dealing with conflicting conclusions (for instance, two UAVs that assign 0.1 and 0.9 as probabilities) is not covered by this approach. Other possibility is to select the belief that assigns maximum probability to the cell. This approach is conservative in the sense that it prefers assigning a true value than a false one to the cell (for instance, it prefers declaring a fire alarm in a cell that not declaring it).

$$l^i_{k,t} \leftarrow \max_{UAV} l^{UAV}_{k,t} \tag{4.5}$$

4.2 Certainty grids for multi-UAV detection and localization of events

As commented, certainty grids and the previous algorithms can be used for missions in which the objective of the fleet is to determine the probability of the presence or absence of objects of a given class in a certain area. For instance, for a fire detection mission, a 2D grid covering the whole area can be defined, each cell storing the probability of having fire. The objective of the UAVs is to update the probabilities of the cells by using their local sensorial data and information received from the companions.

The previous algorithms contain an implicit assumption. The state $\mathbf{X}_t$ in this case is comprised by the status of all the cells of the grid $X_{k,t}$ at time t. The joint posterior $bel(\mathbf{X}_t)$ has to take into account all the possible combinations of values for all the cells, and thus, the full posterior for a grid with L cells should consider 2^L different states (Thrun et al., 2005). Maintaining this posterior is computationally unaffordable. Instead, it is assumed that the state of each cell is independent of the rest, and therefore this posterior can be approximated by the products of its marginals over each cell.

$$bel(\mathbf{X}_t) \approx \prod_k bel(X_{k,t}) \tag{4.6}$$

4.2.1 Likelihood functions

The term $p(\mathbf{z}_{j,t}|X_k)$ is the *likelihood* function of the measurements gathered by UAV j. In the discrete binary case, this likelihood function is characterized by two values: $p(\mathbf{z}_{j,t}|x_k)$, that indicates the probability of having data $\mathbf{z}_{j,t}$ considering that there is an object of the considered class in cell k; and $p(\mathbf{z}_{j,t}|\bar{x}_k)$, that indicates the probability of having data $\mathbf{z}_t$ considering that there is no object of the considered class in cell k .

For each UAV j, the corresponding likelihood function should be specified. This likelihood will take into account the position of the UAV sensors with respect to the map and the geometric characteristics of the distinct sensors (for instance the pin-hole model of the cameras). The latter are obtained through calibration, while the former will be provided by the UAVs. Again, as commented in Chapter 3, the errors in the UAV localization should be taken into account for a correct definition of the likelihood function. As indicated by equation (3.55), the likelihood model is given by:

$$p(\mathbf{z}_{j,t}|x_k) = \int p(\mathbf{z}_{j,t}|x_k, \mathbf{Q}_{j,t}) p(\mathbf{Q}_{j,t}) d\mathbf{Q}_{j,t} \tag{4.7}$$

Equation (4.7) implies a further simplification. Actually, the measurements gathered by UAV j depend conditionally not only on one cell, but at least on all the cells of the grid within the field of view (FOV) $S(j)$ of UAV j. Therefore,

$$p(\mathbf{z}_{j,t}|x_k, \mathbf{Q}_{j,t}) = \sum_{p \in S(j)} \sum_{X_p} p(\mathbf{z}_{j,t}|x_k, X_p, \mathbf{Q}_{j,t}) p(X_p) \tag{4.8}$$

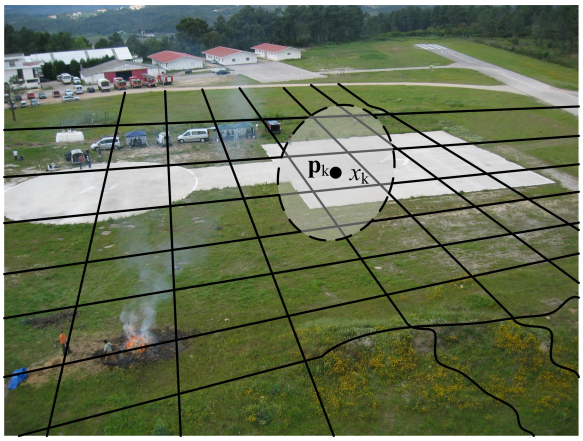

Figure 4.3: Due to the uncertainties in sensor position and the cell resolution, one cell corresponds to a zone on the image plane (represented by the ellipse).

However, as commented, the different cells will be updated independently, and then this dependence will be not considered when computing (4.7).

The rest of the section describes the likelihood functions for several sensors, and for the particular application of fire detection. However, as it will be seen, the likelihood models considered can be easily extended to similar applications.

Likelihood function for the cameras

As presented in Section 3.5, cameras are an important source of information in the fleet of UAVs. Cameras of different modalities can be considered, but, in all the cases, the images gathered are processed so that objects of interest are segmented from the background. For instance, for the case of fire detection, the fire segmentation algorithms of Appendix B are used.

As a result, the measurements provided by the cameras are binary images I that classify the pixels as belonging to the class searched or not. The segmentation algorithms are characterized by the probabilities of correct detection $P_{D,j}$ and false positive $P_{F,j}$ for each UAV j.

In order to determine the likelihood function (4.7), the following should be considered. Each cell k has a position associated, $\mathbf{r}_k$. For a given value of the pose of the sensor $\mathbf{q}_{j,t}$, the center of the cell will correspond to a pixel $\mathbf{m}_{kj,t}$ on the image plane of camera j (if it is within the field of view of that camera). The pixel position is given by eq. (3.57).

If pixel $\mathbf{m}_{kj,t}$ corresponds to a region segmented as fire, then the term $p(\mathbf{z}_{j,t}|x_k, \mathbf{q}_{j,t})$ corresponds to

$$p(\mathbf{z}_{j,t}|x_k, \mathbf{q}_{j,t}) = P_{D,j} \tag{4.9}$$

while if the pixel is classified as background, then the term is given by:

$$p(\mathbf{z}_{j,t}|x_k, \mathbf{q}_{j,t}) = 1 - P_{D,j} \tag{4.10}$$

This last expression allows employing negative information when updating the current belief.

However, the position of the sensor is not known accurately, and, thus, the position of the corresponding pixel is also uncertain. Then, in order to compute the likelihood $p(\mathbf{z}_{j,t}|x_k)$, eq. (4.7) should be integrated for possible values of the uncertain pose of the camera $\mathbf{Q}_{j,t}$. This could be done by sampling values of $\mathbf{Q}_{j,t}$. However, this is computationally expensive, and indeed this should be done for all the cells of the grid that are within the field of view of the camera.

The pixel position is related to the sensor and cell position through the non-linear pin-hole model of eq. (3.57), $\mathbf{M}_{kj,t} = \mathbf{f}(\mathbf{Q}_{j,t}, \mathbf{r}_k)$. Instead of directly solving eq. (4.7), the uncertainties in $\mathbf{Q}_{j,t}$ are propagated into uncertainties on the pixel position $\mathbf{M}_{kj,t}$ corresponding to cell k using the Unscented Transform (UT, see section 3.3). Moreover, in the procedure, the uncertainties in the position $\mathbf{r}_k$ due to the limited resolution of the grid can be taken into account. As a result, each cell k defines a Gaussian distribution on the pixel position $p(\mathbf{M}_{kj,t}) \sim \mathcal{N}(\mathbf{m}_{kj,t}, \mathbf{\Sigma}_{kj,t})$ (see Fig. 4.3). And then, equation (4.7) becomes:

$$p(\mathbf{z}_{j,t}|x_k) = \int p(\mathbf{z}_{j,t}|x_k, \mathbf{Q}_{j,t}) p(\mathbf{Q}_{j,t}) d\mathbf{Q}_{j,t} = \sum_{\mathbf{M}} p(I(\mathbf{M}_{kj,t})|x_k) p(\mathbf{M}_{kj,t}) \tag{4.11}$$

where the sum is taken over a region on the image I determined by the second order moments of the distribution $p(\mathbf{M}_{jk,t})$ (see Fig 4.3).

The same procedure is used to compute the likelihood function $p(\mathbf{z}_{j,t}|\bar{x}_k)$ for the hypotheses $\bar{x}_k$. In this case, equations (4.9) and (4.10) are replaced by:

$$p(\mathbf{z}_{j,t}|\bar{x}_k, \mathbf{Q}_{j,t}) = P_{F,j} \tag{4.12}$$

$$p(\mathbf{z}_{j,t}|\bar{x}_k, \mathbf{Q}_{j,t}) = 1 - P_{F,j} \tag{4.13}$$

To complete the model, the probabilities $P_{D,j}$ and $P_{F,j}$ are modified depending mainly on the relative distance d_{kj} of the cell k, with respect to the mean position of the sensor $\mathbf{q}_j$. Thus:

$$P_{D,j}(\mathbf{r}_k, \mathbf{q}_j) = P_{D,j}\left[1 - (d_{kj}/d_{jmax})^2\right] + 0.5(d_{kj}/d_{jmax})^2 \tag{4.14}$$

$$P_{F,j}(\mathbf{r}_k, \mathbf{q}_j) = P_{F,j}\left[1 - (d_{kj}/d_{jmax})^2\right] + 0.5(d_{kj}/d_{jmax})^2 \tag{4.15}$$

that is, for distances out of the field of view (distances greater than d_{jmax}), $P_{D,j} = P_{F,j}$ (and then, the updating equations do not change the probabilities l_k). This function is plotted in Fig. 4.5, left.

Likelihood function for the fire sensor

In the robot fleet considered in the experiments that will be presented in Chapter 8, one UAV carries on board a sensor that detects the presence of fire in its field of view. This sensor is composed of a fire detector, whose main component is a photodiode set-up to limit its sensibility to the band of $[185, 260]nm$, normally associated to fires (see Fig. 4.4). The data of this sensor can be processed to obtain a binary value indicating if a fire is present or not within the field of view of the sensor.

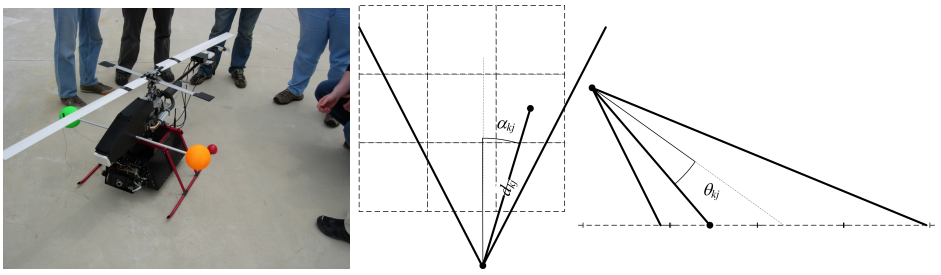

Figure 4.4: Marvin's nose has attached a fire detector sensor. Right: the detection capabilities worsen with the relative distance and angular difference.

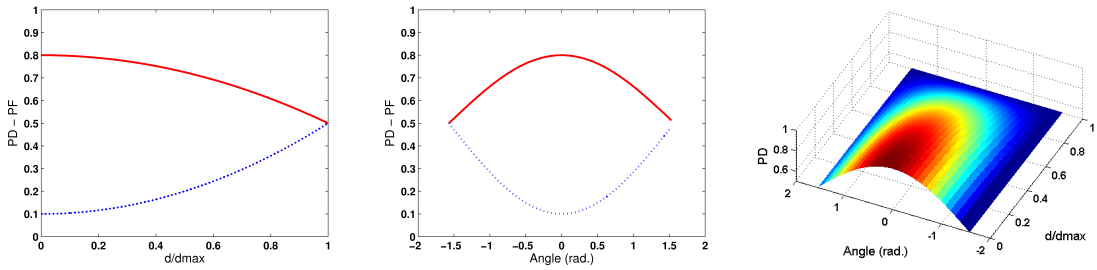

Figure 4.5: Plots of equations $P_{D,j}(\mathbf{r}_k, \mathbf{q}_j)$ and $P_{F,j}(\mathbf{r}_k, \mathbf{q}_j)$. Left, distance component. Middle, angular component. Solid, P_D, dotted, P_F. Right, combined function for $P_{D,j}(\mathbf{r}_k, \mathbf{q}_j)$.

However, the positioning information provided by these sensors is poorer than that provided by the cameras, as they do not provide any bearing information. Also, it is not possible to determine if a measure is due to a big fire far away or a nearby small fire, or if the fire is generated by one or several sources (the data cannot be segmented into contacts).

The functioning of the sensor can be also characterized by probabilities of detection P_D and false positive generation P_F. These probabilities depend on the threshold selected. A higher threshold implies a lower P_F at a cost of worse detection capabilities.

If the sensor detects something, then for the updating of the cells the following values hold:

$$p(\mathbf{z}_{j,t}|x_k, \mathbf{Q}_{j,t}) = P_{D,j}(\mathbf{r}_k, \mathbf{q}_j) \tag{4.16}$$

$$p(\mathbf{z}_{j,t}|\bar{x}_k, \mathbf{Q}_{j,t}) = P_{F,j}(\mathbf{r}_k, \mathbf{q}_j) \tag{4.17}$$

and, in case that the sensor does not detect anything:

$$p(\mathbf{z}_{j,t}|x_k, \mathbf{Q}_{j,t}) = 1 - P_{D,j}(\mathbf{r}_k, \mathbf{q}_j) \tag{4.18}$$

$$p(\mathbf{z}_{j,t}|\bar{x}_k, \mathbf{Q}_{j,t}) = 1 - P_{F,j}(\mathbf{r}_k, \mathbf{q}_j) \tag{4.19}$$

Again, the actual values $P_{D,j}$ and $P_{F,j}$ employed are modified depending on the relative pose of the cell k with respect the sensor j; in this case, considering the distance d_{kj} and the azimuth and

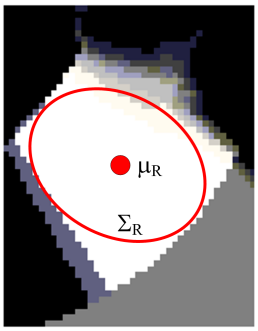

Figure 4.6: It is possible to obtain a Gaussian approximation of the position of regions of high probability.

elevation angles α_{kj}, θ_{kj} (see Fig. 4.4, right), as the fire sensor does not provide bearing information and its detection capabilities decrease with the angle with respect to the sensor axis. Figure 4.5 shows a plot of these functions.

In fact, this model could be used for any other presence sensor that provides binary decisions about the presence or absence of an object of a certain kind within its field of view.

4.2.2 Obtaining measures from the grid

Using the equations described above, the status of the grid is recursively estimated using the data the vehicles are providing. From a Bayesian point of view, the grid represents all the information about the possible alarms at time t. However, in some applications, more specific measures are required. For instance, if a fleet is looking for fire alarms, a control center would expect the position of the potential fire alarm detected, in order to plan a new mission, sending new vehicles to confirm the alarm.

This can be accomplished in various ways. Here, the set of cells of the grid with probabilities over a given threshold is obtained every T seconds. An alarm is raised for each set R of connected cells over this threshold (see Fig. 4.6). The mean position $\boldsymbol{\mu}_R$ of the alarm is computed as the weighted mean of the positions of the cells. Also, it can be obtained an estimation of the uncertainty on the computed position from the second order moments $\boldsymbol{\Sigma}_R$ of the region R.

$$\begin{aligned} \boldsymbol{\mu}_R &= \frac{\sum_{k\subset R} \mathbf{r}_k p(x_k|\mathbf{z}^t)}{\sum_{k\subset R} p(x_k|\mathbf{z}^t)} \\ \boldsymbol{\Sigma}_R &= \frac{\sum_{k\subset R} (\mathbf{r}_k - \boldsymbol{\mu}_R)(\mathbf{r}_k - \boldsymbol{\mu}_R)^T p(x_k|\mathbf{z}^t)}{\sum_{k\subset R} p(x_k|\mathbf{z}^t)} \end{aligned} \tag{4.20}$$

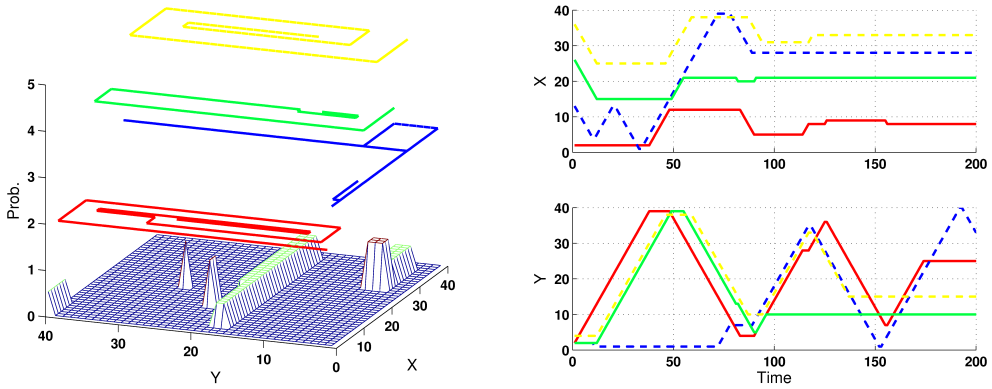

Figure 4.7: In the simulation, 4 UAVs are commanded to look for alarms over the given scenario. The UAVs take off from different places, but no restriction is imposed on their trajectories. Left: final certainty grid and UAVs trajectories. Right: UAVs X and Y trajectories.

4.3 Results

4.3.1 Simulation results

In order to illustrate the previous ideas, first, a simulation setup is developed. In this simulation, a simple model is employed for the motion of the UAVs. Each UAV can decide to stay at the same cell or to move to one of its eight neighbors. A simple greedy algorithm is employed for controlling the UAVs: each UAV decides to move to the cell in which the expected information gain is bigger. This way, the effects of exchanging information on the motion of the fleet will be illustrated.

This information gain is computed as the expected variation on the entropy for the cell. The entropy of each cell k is given by:

$$h_{k,t} = -bel(x_{k,t}) \log[bel(x_{k,t})] - [1 - bel(x_{k,t})] \log[1 - bel(x_{k,t})] \tag{4.21}$$

And the expected entropy after obtaining the measurements is computed as:

$$\begin{aligned} E_{\mathbf{Z}_{t+1}}[h_{k,t+1}(\mathbf{Z}_{t+1})] = h_{k,t+1}(\mathbf{Z}_{t+1} = detect)p(\mathbf{Z}_{t+1} = detect) + \\ + h_{k,t+1}(\mathbf{Z}_{t+1} = no\,detect)p(\mathbf{Z}_{t+1} = no\,detect) \end{aligned} \tag{4.22}$$

where:

$$\begin{aligned} p(\mathbf{Z}_{t+1} = detect) = P_D bel(x_{k,t}) + P_F[1 - bel(x_{k,t})] \\ p(\mathbf{Z}_{t+1} = no\,detect) = (1 - P_D)bel(x_{k,t}) + (1 - P_F)[1 - bel(x_{k,t})] \end{aligned} \tag{4.23}$$

where P_D and P_F define the capabilities of the processing functions and sensors on board the UAV (defined in the previous section). The expected information gain for cell k is then given

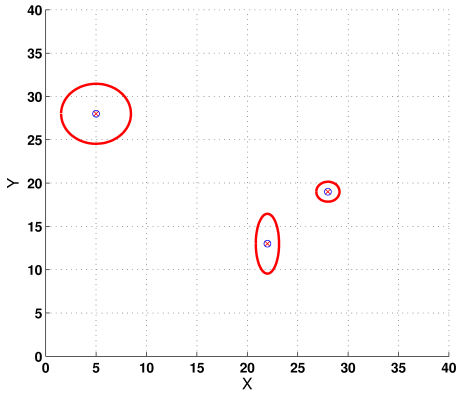

Figure 4.8: Estimated position of the alarms and uncertainty from the final maps.

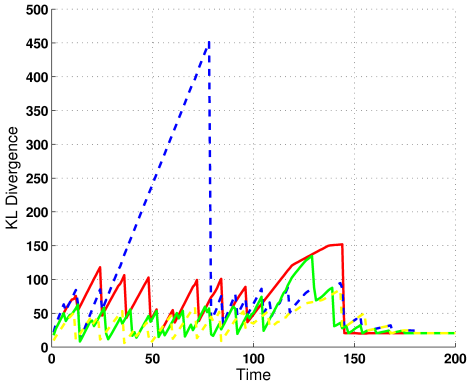

Figure 4.9: Kullback-Leibler divergence of the local estimation (accumulated for all the cells) of each UAV with respect to the estimation on a central node that receives all the information.

by $h_{k,t} - E_{\mathbf{Z}_{t+1}}[h_{k,t+1}(\mathbf{Z}_{t+1})]$ (see Section 2.6). Also, a cost function is included, which penalizes changes in velocity and larger paths. Then, each UAV moves to the cell k^* (within its neighbors) that maximizes:

$$k^* = \arg\max_k \left[\alpha[h_{k,t} - E_{\mathbf{Z}_{t+1}}[h_{k,t+1}(\mathbf{Z}_{t+1})]] - \beta costMove(k)\right] \tag{4.24}$$

where α and β are weights.

In the first experiment, four UAVs are commanded to survey a certain area. Three alarms are situated at certain positions. The grid is initiated so that $bel(x_{k,t}) = 0.5\,\forall k$.

Figure 4.7 shows the final estimated certainty grid along with the trajectories followed by the UAVs. The trajectories are not meaning to be optimal in any sense, but it can be seen that the sharing of information leads to an implicit coordination in the sense that the UAVs tend not to cover places already visited by other UAVs (although there is no explicit restriction for it in the control

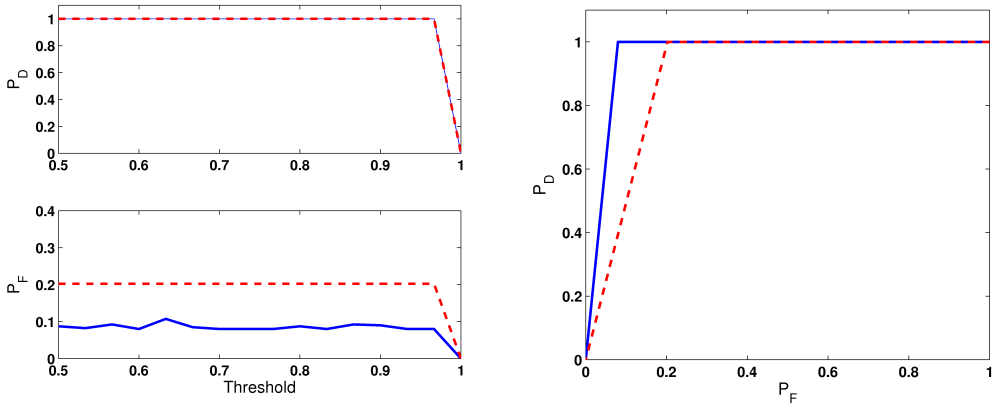

Figure 4.10: Comparison conservative and no conservative fusion rules for Bernouilli variables. Solid: conservative. Dashed: non-conservative. The conservative rule presents a lower false alarm ratio.

algorithms). An estimated position of the alarms and their estimated position employing eq. (4.20) is shown in Fig. 4.8.

A measure of the difference between two different probability distributions is given by the Kullback-Leibler divergence (Kullback and Leibler, 1951) (see Appendix A). For the two different Bernouilli binary and discrete distributions p and q, the KL divergence is defined as:

$$D_{KL}(p\|q) = p(x_t)log\frac{p(x_t)}{q(x_t)} + p(\bar{x}_t)log\frac{p(\bar{x}_t)}{q(\bar{x}_t)} \tag{4.25}$$

Figure 4.9 shows the sum of the KL divergence for all the cells for each UAV with respect to the estimated certainty grid that would be computed by a central node that received all the information. It can be seen how all UAVs tend to have the same belief. The differences with the central node are due to the double counting effect, as in the experiment all the UAVs can communicate with any other in the fleet.

Finally, the standard fusion rule given by Algorithm 4.3 (which is affected by rumor propagation as commented) is compared to the fusion rule described by (4.5), in which each UAV selects the received opinion that assigns the largest probability.

Figure 4.10 shows the averaged probability P_D of detection and the probability of false positives P_F for the fleet in both cases. These ratios are computed as an average of several runs of the same experiment for different threshold values. It can be seen that the conservative rule produces less false alarms than the normal rule for the different thresholds.

4.3.2 Experimental results

Results from one of the experiments carried out in the COMETS project, and described in detail in Chapter 8, are presented here. The experiment shown corresponds to the flight of two UAVs, Marvin and Heliv. Marvin uses locally a fire sensor to estimate the status of the grid (see Fig. 4.4),

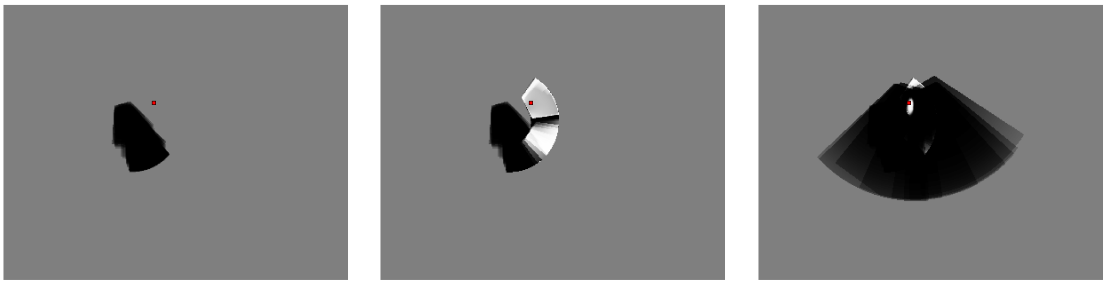

Figure 4.11: The status of the grid at three moments during the mission. The filled square represents the actual position of the fire.

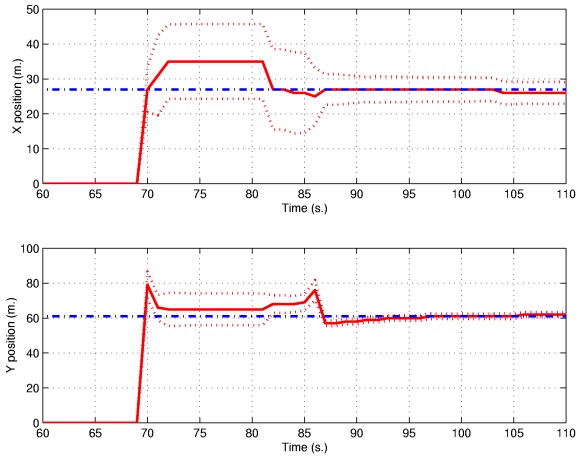

Figure 4.12: Estimated mean position of one of the high probability regions for Marvin. Dotted: estimated variances. Dash-dotted: actual fire position. Around time 70, Marvin detects an alarm with its fire detector (the positioning information is quite limited). Around time 95, Heliv arrives and fuses the information obtained with its IR camera with the previous information obtained by Marvin.

while Heliv employs an infrared camera. In this particular experiment, Marvin is sent to cover a certain region. If a fire alarm is declared, Heliv is sent to the same place to confirm or discard those alarms.

Figure 4.11 shows the evolution of the grid for Marvin in several phases of the experiment. The first image shows the status of the grid after Marvin has flown over a place with no fire, using only the fire sensor. The second image shows how Marvin produces two big high probability blobs on the grid, one due to a false alarm and other due to the actual alarm. Two alarms are generated and Heliv takes off and uses its IR camera over the zone of the possible alarms. The third image shows the fused grid after several images from Heliv are integrated. One of the high probability regions is constrained to a smaller region, which includes the actual position of the fire, while the other is actually eliminated.

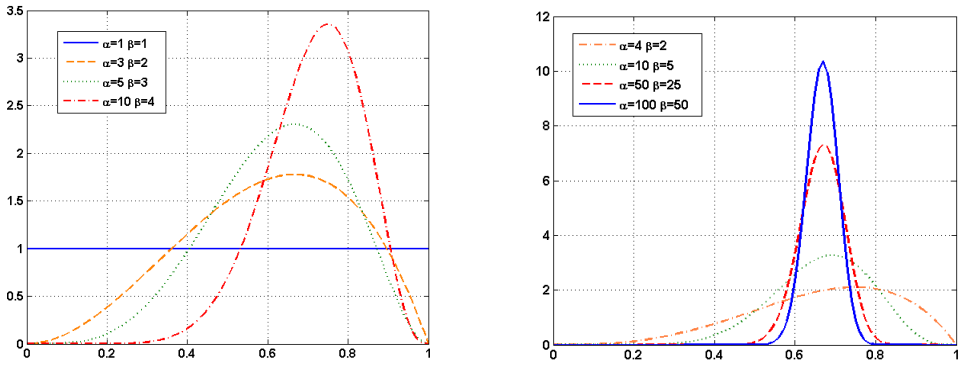

Figure 4.13: The Beta distribution $\mathcal{B}(\alpha, \beta)$ for different values of α and β. Left, Right: different proportions of the values can indicate the same mean with more precision.

Figure 4.12 shows the evolution of the mean position of the high probability region, computed using (4.20), and compares it to the actual fire position. It also shows the estimation on the uncertainty on the computed position.

4.4 The Beta distribution

UAV exploration missions, as those described above, are usually planned beforehand assigning initial probabilities to the cells of the grid, based on information from experts or previous detection missions. Sometimes it is convenient to have a kind of meta-probability measure, or degree of certainty about the assigned Bernouilli probability to each cell, a measure on the evidence support for a mission designer statement. This information cannot be encoded in the probability assigned to each cell.

There are several approaches that try to tackle with this issue, like evidence theory (Shafer, 1976) and possibility theory (Zadeh, 1978), that were commented in Chapter 2. A potential approach within the Bayesian framework is the use of Beta distributions (Bertuccelli and How, 2006). Beta distributions are usually employed as priors for the Bayesian estimation of parameters of Binomial distributions, including Bernouilli distributions. The Beta distribution considers the probability p of occurrence of an event as a random and continuous variable. Observations on the event are used to update the prior given to p.

The pdf of the (univariate) Beta distribution is given by:

$$p(X) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} X^{\alpha-1}(1-X)^{\beta-1} \tag{4.26}$$

for $X \in [0,1]$ and where $\Gamma(a+1)$ is the Gamma function:

$$\Gamma(a+1) = \int_0^\infty t^a e^{-t} dt \tag{4.27}$$

Figure 4.13 shows some plots of the distribution for certain values of α and β. It can be seen how the distribution can adopt a wide variety of shapes. The first two moments of the distribution are given by:

$$\begin{aligned} \mu &= \frac{\alpha}{\alpha+\beta} \\ \sigma &= \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} \end{aligned} \tag{4.28}$$

The mean of the distribution is the Bayes estimator of the underlying Bernouilli coefficient, and the variance can be used as a measure on the evidence supporting this value. Fig. 4.13 right shows several distributions with the same means but different values of α and β, which lead to different degrees of certainty about that coefficient. The values $\alpha = 1$ and $\beta = 1$ correspond to the uniform distribution in the $[0, 1]$ interval.

The idea is then to use the Beta distribution to express not only the probability of having an event in a given cell, but also the robot evidence for this probability.

4.4.1 Bayes filter for Beta distributions

One interesting property is that the Beta distribution is conjugate prior of likelihoods of the Bernouilli form. If X is the probability of occurrence of a certain event, then, if a sensor detects the event, the likelihood of the observation is $p(\mathbf{z}_t|X) = X$ while if the sensor does not detect anything, $p(\mathbf{z}_t|X) = 1 - X$. If the measurements $\mathbf{z}_t$ obtained at time t are a set of m_t detections and n_t non-detections, then

$$p(\mathbf{z}_t|X) = X^{m_t}(1-X)^{n_t} \tag{4.29}$$

Therefore, if the previous belief state is given by $\mathcal{B}(\alpha_{t-1}, \beta_{t-1})$, then, applying the Bayes rule:

$$p(X|\mathbf{z}_t) = \eta X^{m_t}(1-X)^{n_t} X^{\alpha_{t-1}-1}(1-X)^{\beta_{t-1}-1} = \eta X^{\alpha_{t-1}+m_t-1}(1-X)^{\beta_{t-1}+n_t-1} = \mathcal{B}(\alpha_{t-1}+m_t, \beta_{t-1}+n_t) \tag{4.30}$$

Therefore, the updating equations when new information is obtained give as a result a new Beta distribution, with coefficients $\alpha_t = \alpha_{t-1} + m_t$ and $\beta_t = \beta_{t-1} + n_t$.

4.5 Certainty grids employing Beta distributions for multi-UAV detection and localization of events

Following the previous description, a new certainty grid is built so that a Beta distribution is associated to each cell. The mean of the distribution provides the probability of having an event on this cell. The distribution also gives an idea about the support for this affirmation.

In a typical scenario, a mission designer, based on its prior knowledge or as a result of a previous detection mission, assigns to each cell k of the grid a prior Beta distribution with coefficients $\alpha_{k,0}, \beta_{k,0}$

Algorithm 4.4 $\{\alpha_{k,t}, \beta_{k,t}\} \leftarrow$ Beta_Filter$(\{\alpha_{k,t-1}, \beta_{k,t-1}\}, \mathbf{z}_t = \{m_{k,t}, n_{k,t}\})$

1: **for** $k = 1$ to L **do**
2: $\quad \alpha_{k,t} = \alpha_{k,t-1} + m_{k,t}$
3: $\quad \beta_{k,t} = \beta_{k,t-1} + n_{k,t}$
4: **end for**

($\alpha_{k,0} = 1$ and $\beta_{k,0} = 1$ if nothing is known beforehand). During the mission, each UAV will update its belief by using its sensors.

The Bayes filter for one UAV for updating the Beta coefficients of all the cells of the grid is given by Algorithm 4.4. The measurement $\mathbf{z}_t$ consists of $m_{k,t}$ detections and $n_{k,t}$ non-detections for each cell k ($m_{k,t} = 0$ and $n_{k,t} = 0$ for cells outside the field of view of the sensor).

4.5.1 Likelihood for the cameras

As in the previous case, it is assumed that the images provided by the cameras are preprocessed at each time instant, obtaining a binary image I with the segmented objects. As seen on Section 4.2.1, due to the uncertainties on the sensor pose, each cell k of the grid induces a probability distribution $p(\mathbf{M}_{kj})$ on the possible position of the pixels for camera j. This distribution is assumed to be Gaussian and is estimated by using the Unscented Transform.

Therefore, the coefficient $m_{kj,t}$ of detection for a cell and for a particular sensor is given as the mean of pixels classified as fire within the region determined by $p(\mathbf{M}_{kj,t})$, and the coefficient $n_{kj,t}$ the mean of pixels classified as no fire.

$$\begin{aligned} m_{kj,t} &= \sum_{\mathbf{M}_{kj,t}} I(\mathbf{M}_{k,t}) p(\mathbf{M}_{k,t}) \\ n_{kj,t} &= \sum_{\mathbf{M}_{kj,t}} [1 - I(\mathbf{M}_{k,t})] p(\mathbf{M}_{k,t}) \end{aligned} \tag{4.31}$$

where $I(\mathbf{M}_{k,t}) = 1$ for pixels in which a detection is declared, and $I(\mathbf{M}_{k,t}) = 0$ otherwise.

4.5.2 Likelihood for the fire sensor

In the case of the fire detectors or similar sensors, the outputs are binary detections indicating the presence or absence of an event in the field of view. It is not possible to determine how many focuses induce the alarm. Therefore, in the case of detection, the evidence is shared by all the cells within the field of view. Negative information, however, is stronger in this case, as a negative reading indicates that no alarm is within the FOV.

Then, if something is detected, for the cells within the FOV the coefficient is given by:

$$m_{k,t} = \frac{1}{S} \tag{4.32}$$

where S is the area (the number of cells) within the FOV, while $n_{k,t} = 0$, whereas if nothing is detected, then for all the cells within the FOV

$$n_{k,t} = 1 \tag{4.33}$$

while $m_{k,t} = 0$.

4.6 Decentralized estimation employing Beta beliefs

As the state is static, the belief combination formula (2.20) can be applied. If the current belief of UAV i is given by $bel_i(X) = \mathcal{B}(\alpha_t^i, \beta_t^i)$, then, the updated belief state after incorporating the information $bel_j(X) = \mathcal{B}(\alpha_t^j, \beta_t^j)$ received from UAV j is[1]:

$$\begin{aligned} bel_i(X) = \eta X^{\alpha_0-1}(1-X)^{\beta_0-1}\frac{X^{\alpha_t^i+\alpha_0-1}(1-X)^{\beta_t^i-\beta_0-1}}{X^{\alpha_0-1}(1-X)^{\beta_0-1}}\frac{X^{\alpha_t^j+\alpha_0-1}(1-X)^{\beta_t^j-\beta_0-1}}{X^{\alpha_0-1}(1-X)^{\beta_0-1}} = \\ \eta X^{\alpha_t^i+\alpha_t^j-\alpha_0-1}(1-X)^{\beta_t^i+\beta_t^j-\beta_0-1} = \mathcal{B}(\alpha_t^i+\alpha_t^j-\alpha_0, \beta_t^i+\beta_t^j-\beta_0) \end{aligned} \tag{4.34}$$

Therefore, the fusion rule is given by:

$$\begin{aligned} \alpha_t^i &\leftarrow \alpha_t^i + \alpha_t^j - \alpha_0 \\ \beta_t^i &\leftarrow \beta_t^i + \beta_t^j - \beta_0 \end{aligned} \tag{4.35}$$

Also, the recursive fusion rule (2.20) has a simple expression:

$$\begin{aligned} \alpha_t^i &\leftarrow \alpha_t^i + \alpha_t^j - \alpha_{t-1}^j \\ \beta_t^i &\leftarrow \beta_t^i + \beta_t^j - \beta_{t-1}^j \end{aligned} \tag{4.36}$$

4.6.1 Removing common information

If the robot has previously sent information itself to UAV j, then the equations have to be modified consequently to remove also this common information, represented by α_{t-1}^{ij} and β_{t-1}^{ij}. The final fusion rule is given by Algorithm 4.5. However, again this rule can be applied only to the case of a tree-shaped network, as only in this case the common information can be removed employing only local information.

In the case of a more general topology, the unknown common information should be removed. This common information is also a Beta distribution with unknown parameters α^* and β^*. The removal of common information can be determined analytically. From (2.26):

$$\begin{aligned} bel_i(X) \leftarrow \frac{bel_i(X)bel_j(X)}{bel_{ij}(X)} = \eta\frac{\mathcal{B}(\alpha_t^i, \beta_t^i)\mathcal{B}(\alpha_t^j, \beta_t^j)}{\mathcal{B}(\alpha^*, \beta^*)} = \\ \mathcal{B}(\alpha_t^i + \alpha_t^j - \alpha^*, \beta_t^i + \beta_t^j - \beta^*) \end{aligned} \tag{4.37}$$

[1] The cell index is removed for clarity, as the same fusion equation is applied for every cell.

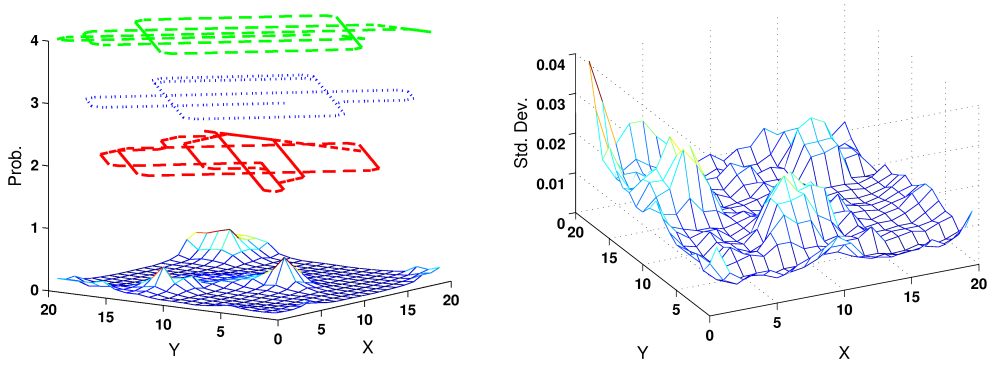

Figure 4.14: In the experiment, 3 UAVs are commanded to look for alarms in a zone. Three alarms in positions $(13, 4)$, $(17, 17)$ and $(7, 10)$ are simulated. Left: mean of the Beta distribution for each cell and UAVs' trajectories. Right: variance of the distribution at each cell.

The main problem is that the values of α^* and β^* are not known in the general decentralized case. However, as the values of α and β are monotonically increasing, then:

$$\begin{aligned} \alpha^* &\in [\alpha_0, \min(\alpha_t^i, \alpha_t^j)] \\ \beta^* &\in [\beta_0, \min(\beta_t^i, \beta_t^j)] \end{aligned} \tag{4.38}$$

that is, the common information will be between the prior (and then, the information gathered by both UAVs is totally independent) and the case in which all the information received was shared previously. A conservative fusion rule can then be devised. In that case, lines 19 and 20 of Algorithm 4.5 are substituted by:

$$\begin{aligned} \alpha_t^i &\leftarrow \max(\alpha_t^i, \alpha_t^j) \\ \beta_t^i &\leftarrow \max(\beta_t^i, \beta_t^j) \end{aligned} \tag{4.39}$$

4.7 Results

Several simulation runs have been tested for the evidence grids employing Beta representations.

In the first experiment, 3 UAVs are commanded to survey a certain zone in which 3 alarms are deployed. Each UAV carries a fire detector that outputs the presence of absence of fire in its FOV. The same greedy exploration algorithm for the control of the UAVs as in the Bernouilli experiments is employed. In this case, the entropy of the Beta distribution is used to compute the expected information gain.

Figure 4.14 shows the trajectories followed by the UAVs and the final estimated probabilities (the mean of the Beta distribution on each cell) for a general decentralized case employing the conservative

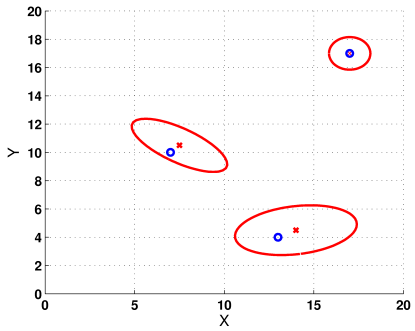

Figure 4.15: Estimated position of the alarms and uncertainty from the final maps.

Figure 4.16: Left: α map. Middle: β map. Right: estimated probability $p = \frac{\alpha}{\alpha+\beta}$.

fusion rule. The figure also shows the associated uncertainty to the estimated Bernouilli probability in each cell.

Regions in which the estimated mean of the Beta distribution is over a threshold are extracted and approximated by a Gaussian distribution by using (4.20). This approximation is shown in Fig. 4.15. Figure 4.16 shows the values of α_k and β_k for each cell, and also the corresponding estimated Bernouilli coefficient.

A second setup considers 4 UAVs. The same experiment is considered twice. In the first run the UAVs cannot communicate, and in the second one, the UAVs can communicate among themselves at certain rates. Fig. 4.17 shows the final map and the trajectories of the UAVs. It can be seen how the sharing of information leads implicitly to a certain coordination of the UAVs, compared to the non-communication case. A detail of the trajectories is given by Fig. 4.18.

Figure 4.19 shows the Kullback-Leibler divergence of each UAV with respect to a central node

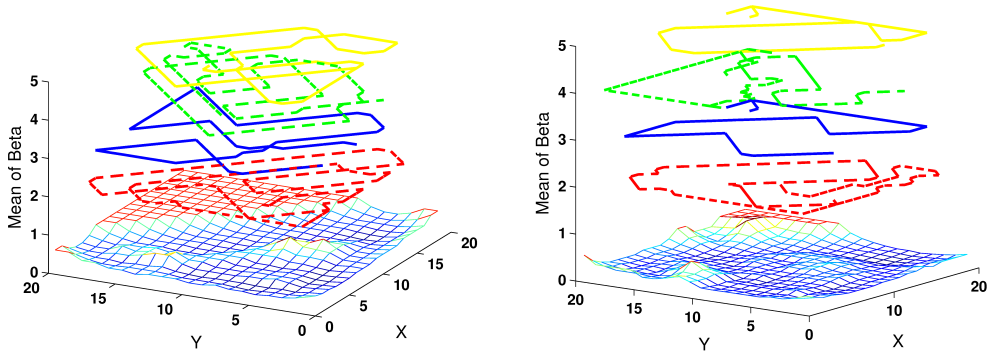

Figure 4.17: Estimated mean and trajectories of the UAVs. Left: non communicating UAVs. Right: communicating UAVs.

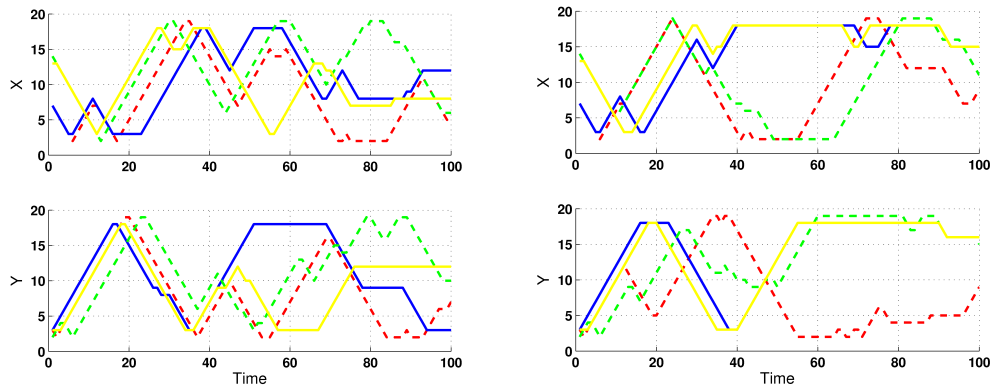

Figure 4.18: Trajectories of the UAVs. Left: non-communicating UAVs. Right: communicating UAVs. It can be seen how communication of beliefs induces an implicit coordination in the trajectories.

that receives all the information at any time, for the non-communicating case and the communicating case. In the latter, it can be seen how, as time goes by, the UAVs tend to have the same belief. As expected, the difference of each UAV's belief with respect to that of the central node is small, although not zero due to the conservative fusion rule.

The same experiment is employed to show the effect of considering the unknown common information. Figure 4.20 shows the KL divergence of the local UAV beliefs with respect to the central belief for the case of the usual fusion rules of eq. (4.36) and the conservative rules of (4.39). It can be seen how the local beliefs will diverge from the actual central one if the rumor propagation is not considered.

Finally, the Beta approach is compared to the Bernouilli-based evidence grids. Figure 4.21 shows the operational curves for the whole fleet for both cases. These curves are obtained by averaging

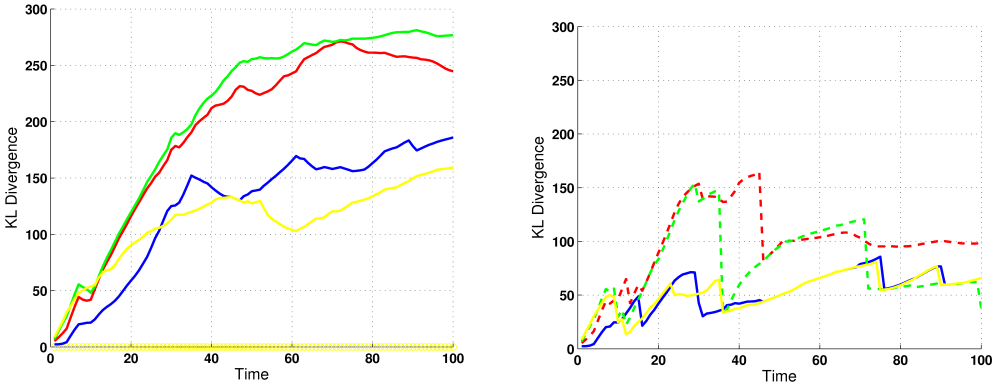

Figure 4.19: Kullback-Leibler divergence with respect to the estimated probabilities in a centralized node. Left: no communicating UAVs. Right: communicating UAVs.

the results obtained in several runs of the same experiment. The Beta-based evidence grids lead to more conservative results, in the sense that less false alarms are usually generated.

One final interesting property of the Beta-based approach is that it does not require to determine a model for the capacities of the different sensors of the fleet. The final estimated grid "learns" the detection capabilities of the whole fleet for each cell.

4.8 Conclusions

This chapter has dealt with cooperative perception employing evidence grids, which is a widespread representation in robotics. Evidence grids based on Bernoulli binary variables have been analyzed and the corresponding decentralized filters shown. It has been pointed out that there is no solution within the Bayesian framework for the problem of a decentralized Bayes filter in the presence of unknown common information. Although the effects can be limited, the rumor propagation effect can produce an increase in the number of false alarms in detection missions. Other fusion rules derived from Consensus Theory as (4.5) can be applied that do not involve the risk of overestimation, at the cost of losing information.

The algorithms have been tested for detection missions in UAVs, but can be extended to other applications, like occupancy grid mapping, etc.

Another approach based on Beta distributions has been presented. Beta distributions are usually employed by staticians as a prior for the estimation of probability of Bernouilli events. The Beta distribution allows to include an idea of the evidence support for the given probability to the cells of the grid. The chapter has shown also how it is possible to derive a decentralized filter for the estimation of the status of the cells.

In the limit, the distributed Beta filter learns the capabilities of the sensors of the fleet for each cell in the grid, without the need of a priori models of the sensors.

Also, the Beta-based Bayes filter allows for a conservative fusion rule of the beliefs. The drawback

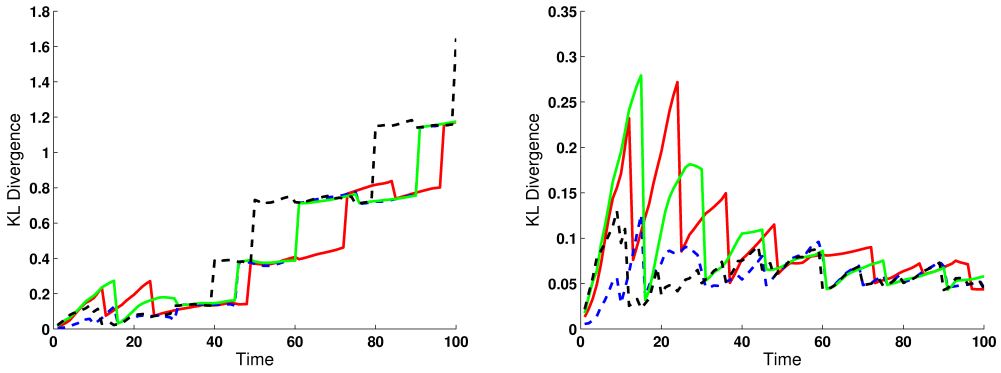

Figure 4.20: Kullback-Leibler divergence with respect to the estimated probabilities in a centralized node for the conservative and no conservative rules. Left: no conservative rule. Right: conservative rule. If no common information is considered, due to rumor propagation, the estimated belief will diverge from the centralized version.

is a higher amount of memory and bandwidth for storing and transmitting the beliefs compared to the Bernouilli case.

Nevertheless, even though they are simple tools for local estimation of static environment information, evidence grids are not very convenient for transmitting information. Chapter 8 will show an hybrid approach in which a local evidence grid is employed to integrate measurements until a certain level of certainty is reached, when a Gaussian approximation is obtained and incorporated into a distributed Information Filter.

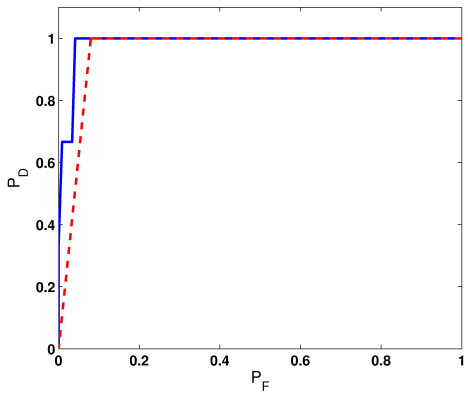

Figure 4.21: Comparison of evidence grids employing Beta distributions and Bernouilli distributions. ROC curve. Solid: Beta-based grid. Dashed: Bernouilli-based grid.

Algorithm 4.5 Beta_Decentralized_Filter(i)

for all k **do**
  $\alpha_{k,0} = 1$/* *Prior* */
  $\beta_{k,0} = 1$
  for all UAV j **do**
    $\alpha_{k,0}^{ij} = 1$
    $\beta_{k,0}^{ij} = 1$
  end for
end for
while true **do**
  if New data $\mathbf{z}_{i,t} = \{m_{k,t}, n_{k,t}\}$ **then** /* *Local information* */
    for all k **do**
      $\alpha_{k,t} = \alpha_{k,t-1} + m_{k,t}$
      $\beta_{k,t} = \beta_{k,t-1} + n_{k,t}$
    end for
  end if
  if New belief from UAV j **then**
    $\Delta\alpha_{k,t}^{j} = \alpha_{k,t}^{j} - \alpha_{k,t-1}^{ij}$/* *Eliminate common information* */
    $\Delta\beta_{k,t}^{j} = \beta_{k,t}^{j} - \beta_{k,t-1}^{ij}$
    $\alpha_{k,t}^{i} \leftarrow \alpha_{k,t}^{i} + \Delta\alpha_{k,t}^{j}$
    $\beta_{k,t}^{i} \leftarrow \beta_{k,t}^{i} + \Delta\beta_{k,t}^{j}$
    $\alpha_{k,t}^{ij} = \alpha_{k,t}^{j}$
    $\beta_{k,t}^{ij} = \beta_{k,t}^{j}$
  end if
  for all UAV j within comm. range **do**
    Send α_t^i, β_t^i to UAV j
    for all k **do**
      $\alpha_{k,t}^{ij} = \alpha_{k,t}^{i}$
      $\beta_{k,t}^{ij} = \beta_{k,t}^{i}$
    end for
  end for
end while

It seems to me what is called for is an exquisite balance between two conflicting needs: the most skeptical scrutiny of all hypotheses that are served up to us and at the same time a great openness to new ideas .

Carl Sagan
The Burden of Skepticism (1987)

Chapter 5

Decentralized perception employing particle filters

In the last years, one of the techniques most used for state estimation under non-linear prediction and measurement functions is particle filtering (Gordon et al., 1993; Doucet et al., 2001). Particle filters employ a non-parametric representation of the belief state different from histograms or grids, which presents some advantages in terms of memory requirements, flexibility and others. The idea in this chapter is to benefit from the advantages of particle filters for cooperative perception.

Therefore, this chapter presents briefly the fundamentals of the particle filters. Then, it develops an approach for decentralized perception employing particle filters. Finally, the technique is applied to vision-based cooperative object detection and localization by a fleet of UAVs.

5.1 Particle filters

5.1.1 Representing the belief state using particles

In particle filters, the belief state at time t is represented by a set of independent and identically distributed (i.i.d.) samples $\{\mathbf{x}_t^{(1)}, \mathbf{x}_t^{(2)}, \ldots, \mathbf{x}_t^{(N)}\}$ according to $p(\mathbf{X}_t|\mathbf{z}^t, \mathbf{u}^t)$. Each sample is called a *particle*. In the same way, the belief state over the full state trajectory $bel(\mathbf{X}^t)$ is expressed as a set of particles $\{\mathbf{x}^{t(1)}, \mathbf{x}^{t(2)}, \ldots, \mathbf{x}^{t(N)}\}$, where each particle $\mathbf{x}^{t(k)}$ represents one potential trajectory $\mathbf{x}^{t(k)} = \{\mathbf{x}_0^{(k)}, \mathbf{x}_1^{(k)}, \ldots, \mathbf{x}_t^{(k)}\}$.

Each particle $\mathbf{x}^{t(k)}$ represents one potential true value for the state trajectory up to time t. The idea behind particle-based representations is related to Monte Carlo integration methods. Given a large number N of particles that are randomly sampled from the true posterior, they represent a good approximation of this posterior, in the sense that, for any given function $\mathbf{f}(\mathbf{X}^t)$:

$$\frac{1}{N}\sum_{k=1}^{N}\mathbf{f}(\mathbf{x}^{t(k)}) \xrightarrow[N\to+\infty]{} \int \mathbf{f}(\mathbf{X}^t)bel(\mathbf{X}^t)d\mathbf{X}^t \tag{5.1}$$

Therefore, the particles, instead of the true distribution, can be used to compute certain quantities

on the state. This can be also interpreted as if the belief state is approximated by the empirical estimate:

$$bel(\mathbf{X}^t) \sim \sum_{k=1}^{N} \delta(\mathbf{x}^{t(k)}) \tag{5.2}$$

where $\delta(\mathbf{x}^{t(k)})$ is the Dirac delta at $\mathbf{x}^{t(k)}$.

5.1.2 Importance sampling

The main problem with a particle representation is that it is usually complicated to draw samples from the posterior $bel(\mathbf{X}^t)$. *Importance sampling* is a method to compute a particle approximation of a complicated distribution, called the *target distribution*, drawing samples from other distribution easier to sample, called the *importance distribution* $\pi(\mathbf{X}^t|\mathbf{z}^t)$ (or *proposal distribution*).

Assuming that $\pi(\mathbf{X}^t|\mathbf{z}^t)$ and $bel(\mathbf{X}^t)$ have the same support, the integral on the right hand side of (5.1) can be expressed as:

$$\int \mathbf{f}(\mathbf{X}^t) bel(\mathbf{X}^t) d\mathbf{X}^t = \frac{\int \mathbf{f}(\mathbf{X}_t)\omega(\mathbf{X}^t)\pi(\mathbf{X}^t|\mathbf{z}^t)d\mathbf{X}^t}{\int \omega(\mathbf{X}^t)\pi(\mathbf{X}^t|\mathbf{z}^t)d\mathbf{X}^t} \tag{5.3}$$

where $\omega(\mathbf{X}^t)$ is called the *importance weight* function,

$$\omega(\mathbf{X}^t) = \frac{\texttt{target}}{\texttt{proposal}} = \frac{bel(\mathbf{X}^t)}{\pi(\mathbf{X}^t|\mathbf{z}^t)} \tag{5.4}$$

If one obtains N i.i.d. samples $\mathbf{x}^{t(k)}$ from $\pi(\mathbf{X}^t|\mathbf{z}^t)$, an estimation of the previous integral is given by:

$$\int \mathbf{f}(\mathbf{X}^t) bel(\mathbf{X}^t) d\mathbf{X}^t \approx \frac{\frac{1}{N}\sum_{k=1}^{N} \mathbf{f}(\mathbf{x}^{t(k)})\omega(\mathbf{x}^{t(k)})}{\frac{1}{N}\sum_{k=1}^{N} \omega(\mathbf{x}^{t(k)})} = \frac{1}{N}\sum_{k=1}^{N} \mathbf{f}(\mathbf{x}^{t(k)})\tilde{\omega}^{t(k)} \tag{5.5}$$

where

$$\tilde{\omega}^{t(k)} = \frac{\omega(\mathbf{x}^{t(k)})}{\sum_{j=1}^{N} \omega(\mathbf{x}^{t(j)})} \tag{5.6}$$

Equation (5.5) can be also interpreted as if the belief state was approximated by the set of *weighted particles* $\{\langle \mathbf{x}^{t(1)}, \tilde{\omega}^{t(1)} \rangle, \langle \mathbf{x}^{t(2)}, \tilde{\omega}^{t(2)} \rangle, \ldots, \langle \mathbf{x}^{t(N)}, \tilde{\omega}^{t(N)} \rangle\}$, so that:

$$bel(\mathbf{X}^t) \sim \sum_{k=1}^{N} \tilde{\omega}^{t(k)} \delta(\mathbf{x}^{t(k)}) \tag{5.7}$$

In this case, the main issues are to select an adequate importance distribution and to compute the corresponding weights of the particles.

Algorithm 5.1 $(\{\mathbf{x}^{t(k)}; k = 1, \ldots, N\}) \leftarrow$ Particle_filter$(\{\mathbf{x}^{t-1(k)}; k = 1, \ldots, N\}, \mathbf{z}_t)$

1: $\tilde{X}_t = \{\emptyset\}$
2: **for** $k = 1$ to N **do**
3: sample $\mathbf{x}_t^{(k)} \sim \pi(\mathbf{X}_t|\mathbf{x}_{t-1}^{(k)}, \mathbf{z}_t)$/* *Prediction* */
4: $\tilde{\omega}_t^{(k)} = \frac{p(\mathbf{z}_t|\mathbf{x}_t^{(k)})p(\mathbf{x}_t^{(k)}|\mathbf{x}_{t-1}^{(k)})}{\pi(\mathbf{x}_t^{(k)}|\mathbf{x}_{t-1}^{(k)}, \mathbf{z}_t)}$/* *Updating* */
5: Set $\mathbf{x}^{t(k)} = \{\mathbf{x}_t^{(k)}, \mathbf{x}^{t-1(k)}\}$
6: Add $\langle \mathbf{x}^{t(k)}, \tilde{\omega}_t^{(k)} \rangle$ to $\tilde{X}_t$
7: **end for**
8: Normalize weights $\{\tilde{\omega}_t^{(k)}\}, k = 1, \ldots, N$ in $\tilde{X}_t$
9: Resample with replacement N particles from $\tilde{X}_t$ with probability proportional to the weights $\tilde{\omega}_t^{(k)}$

5.1.3 Sequential importance sampling with resampling

For the actual operation of a robot, it is interesting to be able to construct the particle trajectories recursively; i.e., to compute the particle representation at time t $\{\mathbf{x}^{t(k)}, k = 1, \ldots, N\}$ from and without affecting the previous trajectories $\{\mathbf{x}^{t-1(k)}, k = 1, \ldots, N\}$.

This can be accomplished if the importance function can be factorized as:

$$\pi(\mathbf{X}^t|\mathbf{z}^t) = \pi(\mathbf{X}_t|\mathbf{X}^{t-1}, \mathbf{z}_t)\pi(\mathbf{X}^{t-1}|\mathbf{z}^{t-1}) \tag{5.8}$$

i.e., the function $\pi(\mathbf{X}^{t-1}|\mathbf{z}^{t-1})$ is the marginal at time $t-1$ of $\pi(\mathbf{X}^t|\mathbf{z}^t)$. In this case, each time step the new sample set $\{\langle \mathbf{x}^{t(k)}, \tilde{\omega}^{t(k)} \rangle\}$ is obtained from the set $\{\langle \mathbf{x}^{t-1(k)}, \tilde{\omega}^{t-1(k)} \rangle\}$, sampling particles from the proposal $\mathbf{x}_t^{(k)} \sim \pi(\mathbf{X}_t|\mathbf{x}_{t-1}^{(k)}, \mathbf{z}_t)$ and updating accordingly the weights by considering the new information. As commented, the particular form of the proposal distribution allows updating the weights recursively:

$$\tilde{\omega}_t^{(k)} = \frac{bel(\mathbf{x}^{t(k)})}{\pi(\mathbf{x}^{t(k)}|\mathbf{z}^t)} = \frac{\eta p(\mathbf{z}_t|\mathbf{x}_t^{(k)})p(\mathbf{x}_t^{(k)}|\mathbf{x}_{t-1}^{(k)})bel(\mathbf{x}^{t-1(k)})}{\pi(\mathbf{x}_t^{(k)}|\mathbf{x}_{t-1}^{(k)}, \mathbf{z}_t)\pi(\mathbf{x}^{t-1(k)}|\mathbf{z}^{t-1})} \propto \tilde{\omega}^{t-1(k)} \frac{p(\mathbf{z}_t|\mathbf{x}_t^{(k)})p(\mathbf{x}_t^{(k)}|\mathbf{x}_{t-1}^{(k)})}{\pi(\mathbf{x}_t^{(k)}|\mathbf{x}_{t-1}^{(k)}, \mathbf{z}_t)} \tag{5.9}$$

From the derivation of the previous equations, it can be seen that if the particles $\{\langle \mathbf{x}^{t(k)}, \tilde{\omega}_t^{(k)} \rangle, k = 1, \ldots, N\}$ are distributed according to $bel(\mathbf{X}^t)$, then $\{\langle \mathbf{x}_t^{(k)}, \tilde{\omega}_t^{(k)} \rangle, k = 1, \ldots, N\}$ are distributed according to the marginal $bel(\mathbf{X}_t)$ (corresponding to the filtering distribution).

The main drawback of sequential importance sampling is that, as time increases, many particles will end up in regions of the state space of low probability; therefore, more new particles would be needed to maintain an accurate approximation of the belief. One way to overcome this problem is to include a resampling step, in which particles with low weights are eliminated, while particles having high importance weights are replicated (Gordon et al., 1993). In order to do this, one option is to draw with replacement samples from $\{\mathbf{x}^{t(k)}, i = 1, \ldots, N\}$, with probability proportional to each particle weight $\tilde{\omega}_t^{(k)}$. This way, a new particle set $\{\mathbf{x}^{t(k)}, k = 1, \ldots, N\}$ with uniform weights is obtained that still represents the belief state.

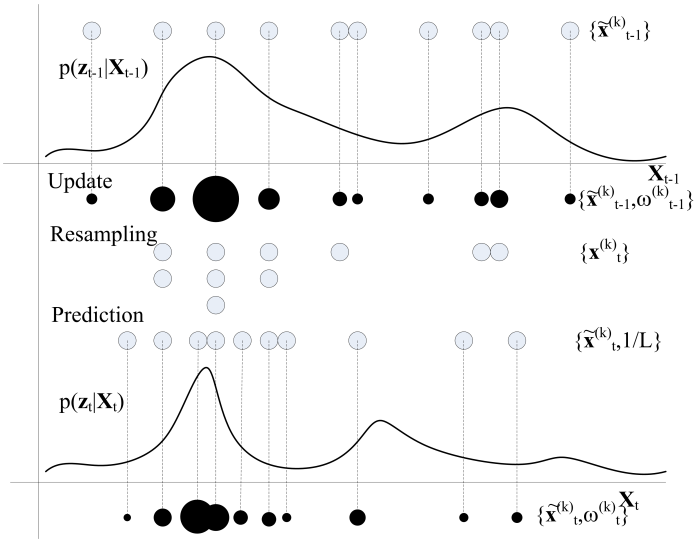

Figure 5.1: Scheme of the particle filter in Algorithm 5.1 (adapted from (Doucet et al., 2001)).

All the previous ideas are considered in the basic particle filter (also called the bootstrap filter) for belief state estimation, described in Algorithm 5.1. The filter computes the particles for the trajectory up to time t from the trajectories up to time $t-1$ and the current measurements. It does so by computing a new set of particles $\mathbf{x}_t^{(k)}$ (and its corresponding weights) and appending it to the previous trajectory, obtaining a representation of $bel(\mathbf{X}^t)$. At the same time, the particles $\mathbf{x}_t^{(k)}$ are a representation of $bel(\mathbf{X}_t)$.

Figure 5.1 shows a graphical description of the filter for the particular case in which the prediction function is used as importance function $\pi(\mathbf{X}^t) = p(\mathbf{X}_0^t) = p(\mathbf{X}_0)\prod_{\tau=1}^{\tau=t} p(\mathbf{X}_\tau|\mathbf{X}_{\tau-1})$. In this case, the importance function can be factored as $\pi(\mathbf{X}^t) = p(\mathbf{X}_t|\mathbf{X}_{t-1})\pi(\mathbf{X}^{t-1})$. The weight updating stage (5.9) adopts a particularly simple form:

$$\tilde{\omega}_t^{(k)} \propto \tilde{\omega}^{t-1(k)} p(\mathbf{z}_t|\mathbf{x}_t^{(k)}) \tag{5.10}$$

that is, each particle weight is updated by the likelihood of the data for the particular state $\mathbf{x}_t^{(k)}$. The figure shows how from a weighted set at time $t-1$, $\{\langle \mathbf{x}_{t-1}^{(k)}, \tilde{\omega}_{t-1}^{(k)} \rangle\}$, the resampling step generates a new set of particles with uniform weights, by copying several times particles with high weights while eliminating those with low weights. Then, the prediction step will produce different predicted states at time t. The predicted particles will be weighted according to the likelihood of the measurements obtained.

One key aspect is the resampling algorithm. As pointed out in (Thrun et al., 2005), while resampling allows to vary the resolution of the particle representation, focusing on zones of the state space with high likelihood, at the same time the variance of the particle set as an estimator increases. It is important to control this variance. There are two additional steps considered for overcoming this problem.

First, the resampling stage is not always performed. One potential approach is to resample only if the variances of the weights is below a certain threshold. This is called the effective number of particles $N_{eff} = [\sum_k \tilde{\omega}^{(k)2}]^{-1}$.

Also, there are several resampling algorithms that can be used. Some of them try to control the variance of the resulting particles, as the low variance sampler described in (Thrun et al., 2005).

5.2 Decentralized perception with particle filters

The next question is if the previous filter can be decentralized among the robots of the fleet. Each robot will be running a local particle filter for estimating its belief $bel_i(\mathbf{X}_t)$. Therefore, at time t, the belief or robot i is given by a set of particles:

$$bel_i(\mathbf{X}_t) \sim \{\mathbf{x}_{i,t}^{(k)}\} \tag{5.11}$$

$$bel_i(\mathbf{X}^t) \sim \{\mathbf{x}_i^{t(k)}\} \tag{5.12}$$

where $\mathbf{x}_i^{t(k)} = \{\mathbf{x}_{i,t}^{(k)}, \mathbf{x}_{i,t-1}^{(k)}, \ldots, \mathbf{x}_{i,0}^{(k)}\}$ is the belief over the state trajectory.

In the hypothetical case of a central node that receives all the particles, Section 2.4.2 has shown that, if all robots share the same prediction functions and prior, then, the global belief can be obtained as:

$$bel_m(\mathbf{X}^t) = \eta p(\mathbf{X}_0^t) \prod_{i=1}^{M} \frac{bel_i(\mathbf{X}^t)}{p(\mathbf{X}_0^t)} \tag{5.13}$$

with $p(\mathbf{X}_0^t) = p(\mathbf{X}_0) \prod_{\tau=1}^{\tau=t} p(\mathbf{X}_\tau | \mathbf{X}_{\tau-1})$.

In the case of a particle representation, each UAV i sends its particles $\{\langle \mathbf{x}_i^{t(k)}, \tilde{\omega}_i^{t(k)} \rangle, k = 1, \ldots, N\}$ to the central node. If the M UAVs consider the same importance function $\pi(\mathbf{X}^t) = p(\mathbf{X}_0^t)$ and share the same particle trajectories $\mathbf{x}^{t(k)}$, then the new weight for each trajectory when incorporating information from all the robots would be

$$\tilde{\omega}^{m,t(k)} = \frac{bel_m(\mathbf{x}^{t(k)})}{\pi(\mathbf{x}^{t(k)})} = \eta \frac{p(\mathbf{x}^{t(k)})}{p(\mathbf{x}^{t(k)})} \prod_{i=1}^{M} \frac{bel_i(\mathbf{x}^{t(k)})}{p(\mathbf{x}^{t(k)})} \propto \prod_{i=1}^{M} \tilde{\omega}_i^{t(k)} \tag{5.14}$$

where $\tilde{\omega}_i^{t(k)}$ is the weight given by UAV i to particle k.

However, even considering the same prior, given the random nature of the prediction and resampling steps (steps 3 and 9 of Algorithm 5.1), each UAV will maintain a different particle set $\mathbf{x}_i^{t(k)}$ and, therefore, the previous products are not well defined.

In order to overcome this problem, Coates (2004) employs *synchronized* particle filters, in which the random number generators are adjusted so each local particle filter generates the same trajectories. There are other approaches for distributed particle filters, like the one presented in (Sheng et al., 2005), in which the different local filters are run in parallel and send their beliefs to a fusion centre. In this case, no synchronization is needed. Before sending to the fusion centre, the particles

are approximated by a continuous distribution: Gaussian Mixture Models, an idea that will be revisited later. In the field of distributed computing, Bolic et al. (2005) proposes the use of a central unit that arbitrates the resampling steps and the flow of particles between the local filters.

Nevertheless, the interest is to develop decentralized belief sharing employing particles, with no need of a central node or intensive communication among the local filters. Rosencrantz et al. (2003) propose a decentralized approach employing distributed particle filters. In this case, the information sharing is based on a query-response protocol, in which a robot queries a companion sending its particle trajectories: the companion answers with the local measurement most informative for the querying robot (the measurement that changes its belief most). In the paper, the communication overload that involves sharing all the particle trajectories is pointed out, and thus only a small set of all the particles are included in the query.

In the approach followed here, each robot shares its particles with its companions. As commented before, communicating the full trajectories is unaffordable in the general case, due to bandwidth restrictions. A suboptimal approach would be only to share the marginal $\mathbf{x}_t^{(k)}$, that is, the particle representation of the belief at time t. The cost is, as commented in Section 2.4.2, that some information is lost in the dynamic-state case, and one cannot recover the same belief that would be recovered in a central node with all information.

The algorithm for fusing the received information is based on the following idea: the particle set received from the remote UAV is incorporated temporally into the local filter. The fusion rule consists of re-weighting adequately the particles such that they are distributed according to the centralized fused belief state $bel_m(\mathbf{X}_t)$.

Recalling equations and expressions of Section 2.5, the fusion rule when UAV i receives information from UAV j is:

$$bel_{i,m}(\mathbf{X}_t) = \eta \frac{bel_i(\mathbf{X}_t) bel_j(\mathbf{X}_t)}{bel_{ij}(\mathbf{X}_t)} \tag{5.15}$$

where $bel_{ij}(\mathbf{X}_t)$ represents all the common information shared by i and j (for instance, due to past received information from UAV j, $bel_j(\mathbf{X}_{t-1})$, information sent by UAV i to UAV j and other common information due to double counting, etc.).

Considering an un-weighted set of particles, the received particles from UAV j are distributed according to $bel_j(\mathbf{X}_t)$ and therefore should be weighted as:

$$\omega_{j,t}^{(k)} = \frac{bel_{i,m}(\mathbf{x}_{j,t}^{(k)})}{bel_j(\mathbf{x}_{j,t}^{(k)})} = \eta \frac{bel_i(\mathbf{x}_{j,t}^{(k)})}{bel_{ij}(\mathbf{x}_{j,t}^{(k)})} \tag{5.16}$$

while the weight for the local particles i is computed as:

$$\omega_{i,t}^{(k)} = \eta \frac{bel_j(\mathbf{x}_{i,t}^{(k)})}{bel_{ij}(\mathbf{x}_{i,t}^{(k)})} \tag{5.17}$$

Similar expressions can be obtained for the case that the weighted sets of particles $\{\langle \mathbf{x}_{i,t}^{(k)}, \tilde{\omega}_{i,t}^{(k)} \rangle\}$ and $\{\langle \mathbf{x}_{j,t}^{(k)}, \tilde{\omega}_{j,t}^{(k)} \rangle\}$ are considered.

Algorithm 5.2 $(\{\mathbf{x}_{i,t}^{(k)}\}_{k=1,\ldots,N_i}) \leftarrow$fuse_particles$(\{\mathbf{x}_{j,t}^{(k)}\}_{k=1,\ldots,N_j}, \{\mathbf{x}_{i,t}^{(k)}\}_{k=1,\ldots,N_i})$

1: $\tilde{X}_t = \{\emptyset\}$
2: $bel_j(\mathbf{X}_t) \leftarrow$ **Approximate_belief**$(\{\mathbf{x}_{j,t}^{(k)}\})$
3: $bel_i(\mathbf{X}_t) \leftarrow$ **Approximate_belief**$(\{\mathbf{x}_{i,t}^{(k)}\})$
4: **for** $k = 1$ to N_j **do**
5: $\quad \omega_{j,t}^{(k)} = bel_i(\mathbf{x}_{j,t}^{(k)})/bel_{ij}(\mathbf{x}_{j,t}^{(k)})$
6: $\quad$ Add $\langle \mathbf{x}_{j,t}^{(k)}, \omega_{j,t}^{(k)} \rangle$ to $\tilde{X}_t$
7: **end for**
8: **for** $k = 1$ to N_i **do**
9: $\quad \omega_{i,t}^{(k)} = bel_j(\mathbf{x}_{i,t}^{(k)})/bel_{ij}(\mathbf{x}_{i,t}^{(k)})$
10: $\quad$ Add $\langle \mathbf{x}_{i,t}^{(k)}, \omega_{i,t}^{(k)} \rangle$ to $\tilde{X}_t$
11: **end for**
12: $bel_{ij}(\mathbf{X}_t) \leftarrow bel_j(\mathbf{X}_t)$
13: Normalize weights $\{\omega_t^{(k)}\}, k = 1, \ldots, N_j + N_i$
14: Resample with replacement N_i particles $\{\mathbf{x}_{i,t}^{(k)}; k = 1, \ldots, N_i\}$ from the set $\tilde{X}_t$ according to the weights $\{\tilde{\omega}_t^{(k)}; k = 1, \ldots, N_j + N_i\}$

The main problem is that $bel_i(\mathbf{x}_{j,t}^{(k)})$ and $bel_j(\mathbf{x}_{i,t}^{(k)})$ are not known, as $bel_i(\mathbf{X}_t)$ and $bel_j(\mathbf{X}_t)$ are represented as sets of particles. In order to perform the previous operations, it is needed to recover a functional expression for the beliefs $bel_j(\mathbf{X}_t)$, $bel_i(\mathbf{X}_t)$ and $bel_{ij}(\mathbf{X}_t)$.

The final equations for fusing the received particles are described in Algorithm 5.2. As commented in Section 2.5, after the particles from UAV j are received the belief $bel_j(\mathbf{X}_t)$ is stored as the common information for the next step. Although not shown in the Algorithm, in the same way, any time that UAV i sends information to UAV j, this information is stored as the common information.

Next section presents some methods that can be employed for recovering a functional expression of the belief from the set of particles (the **Approximate_belief** function in the previous algorithm).

5.2.1 Density estimation from the particle set

An approximation that is quite fast is to approximate $bel_j(\mathbf{x}_{i,t}^{(k)})$ and $bel_i(\mathbf{x}_{i,t}^{(k)})$ by uniform distributions, and to use these uniform distributions to reweight the particles. However, this approximation is too crude. Some authors employ grid representations for approximating the density represented by the particles. For instance, Fox et al. (2001) propose the use of density trees to obtain a variable-resolution grid approximation to the belief in the context of multi-robot localization.

Other options employed by some authors are mixtures, like Gaussian Mixture Models (GMMs). A GMM is expressed as:

$$bel(\mathbf{X}_t) = \sum_k \alpha_k \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{5.18}$$

with $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ the mean and covariance matrix of each component of the mixture, and α_k the weights of each Gaussian in the mixture ($\sum_k \alpha_k = 1$).

There are several methods to obtain a GMM from a particle representation. Expectation-Maximization is employed by Sheng et al. (2005), although it is costly computationally speaking. One interesting option due to their analytical properties is to use a Kernel regularisation or Kernel density estimator (Musso et al., 2001; Forssén, 2004; Ong et al., 2005b). A kernel density estimator estimates the value of the belief at any point $\mathbf{X}_t$ as:

$$bel(\mathbf{X}_t) = \frac{1}{Nh^d}\sum_{k=1}^{N} K(\frac{\mathbf{X}_t - \mathbf{x}_t^{(k)}}{h}) \tag{5.19}$$

where $K(\mathbf{X}_t)$ is the kernel, h is a scaling parameter called the Kernel bandwidth, N is the number of particles and d is the dimension of the state. Requiring that:

$$H(\mathbf{X}_t) \geq \mathbf{0} \texttt{ and } \int H(\mathbf{X}_t)d\mathbf{X}_t = 1 \texttt{ for } H(\mathbf{X}_t) = \frac{1}{h^d}K(\frac{\mathbf{X}_t}{h}) \tag{5.20}$$

then the estimated belief is a correct pdf, and can be expressed as:

$$bel(\mathbf{X}_t) = \frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_t - \mathbf{x}_t^{(k)}) \tag{5.21}$$

that is, the estimated pdf is a sample average of the shifted kernels $H(\mathbf{X}_t - \mathbf{x}_t^{(k)})$. In the limit:

$$\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N} H(\mathbf{X}_t - \mathbf{x}_t^{(k)}) = E_{\mathbf{x}_t}\{H(\mathbf{X}_t - \mathbf{x}_t)\} = \int bel(\mathbf{X}_\tau)H(\mathbf{X}_t - \mathbf{X}_\tau)d\mathbf{X}_\tau = (bel * H)(\mathbf{X}_t) \tag{5.22}$$

where $*$ is the convolution operator. That is, in the limit, this estimation is a low-pass filtered version or regularisation of the belief state $bel(\mathbf{X}_t)$. The kernel bandwidth h is selected in order to minimize the error between the kernel estimation and the particle-based estimation (Musso et al., 2001).

Here a Gaussian Kernel is considered, as proposed by Ong et al. (2005b). As a result, the belief state is approximated by a GMM. If the covariance matrix for the particle distribution is $\mathbf{\Sigma}$, it can be seen that the final kernels are:

$$H(\mathbf{X}_t) = \frac{1}{h^d|\mathbf{\Sigma}|^{0.5}}e^{(-\frac{1}{2h^2}\mathbf{X}_t^T\mathbf{\Sigma}^{-1}\mathbf{X}_t)} \tag{5.23}$$

and the optimal bandwidth is:

$$h_{opt} = (\frac{4}{d+2})^{\frac{1}{d+4}}N^{-\frac{1}{d+4}} \tag{5.24}$$

Therefore, the kernel estimation is given by

$$bel_j(\mathbf{X}_t) = \sum_k \omega^{(k)}\frac{1}{h^d}\mathcal{N}(\mathbf{x}_t^{(k)}, h\mathbf{\Sigma}) \tag{5.25}$$

The number of GMM components obtained is very high (equal to the number of particles). Therefore, it should be reduced in order to have a tractable representation if further operations were

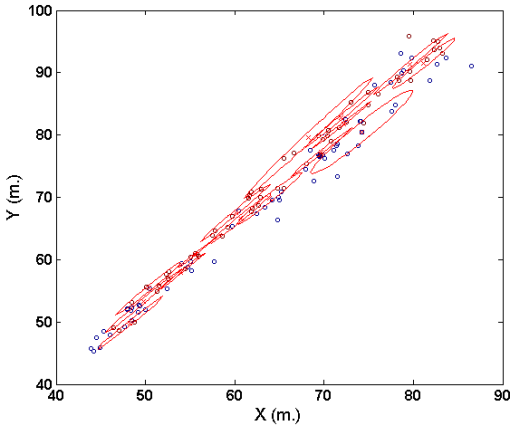

Figure 5.2: Gaussian Mixture Model for a set of particles representing the 2D position of an object.

required. There are several options, like the k-means algorithm (Sheng et al., 2005). Figure 5.2 shows an example of the GMM approximation of a particle set.

5.2.2 Conservative fusion

The previous filter ensures eliminating the common information only in the case that there are no multiple paths for the same information. However, there is no an analytical rule for conservative fusion of the particles for general topologies.

Upcroft et al. (2005) have proposed a conservative rule for mixing GMMs, that could be used here to fuse first the GMMs and then sample from the resultant distribution. Nevertheless, there is still no prove for the consistency of the proposed method.

Moreover, the computational complexity of the GMM approximation is high. Therefore, a different fusion approach based on Consensus Theory is also considered, and will be compared with the fusion based on the functional recovery.

5.2.3 Fusing particles using Consensus Theory

A different option for fusing the particles is to use Consensus Theory, briefly described in Chapter 2. One of the methods employed in this theory is the Linear Opinion Pool (LOP), in which the consensus belief is given by a weighted sum of the expert beliefs:

$$bel_i(\mathbf{X}_t) = \sum_j \alpha_j bel_j(\mathbf{X}_t) \tag{5.26}$$

with $\sum_j \alpha_j = 1$. By using a LOP, the different particle sets are weighted with a factor α_j representing how much confidence the local UAV has in the received information.

Therefore, after fusion, the new belief state for UAV i is approximated by:

Algorithm 5.3 $(\{\mathbf{x}_{i,t}^{(k)}\}_{k=1,\ldots,N_i} \leftarrow$fuse_particles_LOP$(\{\mathbf{x}_{j,t}^{(k)}\}_{k=1,\ldots,N_j}, \{\mathbf{x}_{i,t}^{(k)}\}_{k=1,\ldots,N_i})$

1: $X_t = \{\emptyset\}$, $\tilde{X}_t = \{\emptyset\}$
2: **for** $k = 1$ to N_j **do**
3: $\quad \omega_{j,t}^{(k)} = \frac{1}{N_j}\alpha_j$
4: $\quad$ Add $\langle \mathbf{x}_{j,t}^{(k)}, \omega_{j,t}^{(k)} \rangle$ to $\tilde{X}_t$
5: **end for**
6: **for** $k = 1$ to N_i **do**
7: $\quad \omega_{i,t}^{(k)} = \frac{1}{N_i}\alpha_i$
8: $\quad$ Add $\langle \mathbf{x}_{i,t}^{(k)}, \omega_{i,t}^{(k)} \rangle$ to $\tilde{X}_t$
9: **end for**
10: Normalize weights $\{\omega_t^{(k)}\}, k = 1, \ldots, N_j + N_i$
11: Resample with replacement N_i particles $\{\mathbf{x}_{i,t}^{(k)}; k = 1, \ldots, N_i\}$ from the set $\tilde{X}_t$ according to the weights $\{\tilde{\omega}_t^{(k)}; k = 1, \ldots, N_j + N_i\}$

$$bel_i(\mathbf{X}_t) \sim \alpha_i \frac{1}{N_i} \sum_{k=1}^{N_i} \delta(\mathbf{x}_{i,t}^{(k)}) + \alpha_j \frac{1}{N_j} \sum_{k=1}^{N_j} \delta(\mathbf{x}_{j,t}^{(k)}) \tag{5.27}$$

where $\alpha_i + \alpha_j = 1$ and $\{\mathbf{x}_{j,t}^{(k)}, k = 1, \ldots, N_j\}$ is the particle set received from UAV j. This new set is then employed locally by UAV i during one iteration, and, in the new resampling stage, the number of particles can be reduced to the original number. The final fusion rule is given by Algorithm 5.3.

In this case, instead of reweighting the particles, the UAVs reach a consensus, the fused belief is the sum of the individual beliefs. In this fusion rule, just after receiving the particle set, the variance of the particles can increase. It is the resampling step which produces the reduction in this variance. The received particle set will add a higher density of particles around the "true" value of the state, and therefore the resampling step will leave a higher density of particles around this value. The particles received with very low likelihoods will be promptly removed after a few new measurements.

Therefore, the drawback is that the information gain is less than in the previous case. The advantage is, on one hand, that this approach is much simpler than the previous one, and the computational power required is less. But also the risk of non-consistent estimations is lower.

5.3 Cooperative detection and localization employing particle filters

In order to illustrate the information fusion using particles, the particle representation will be employed for the problem of object localization by using cameras, defined in previous Chapters.

As it will be seen, the particle filter allows to relax some of the assumptions considered before. For instance, cameras only provide bearing measurements, and therefore it is not possible to determine the position of one object given only one view of it, if no further assumptions are made. Also, the

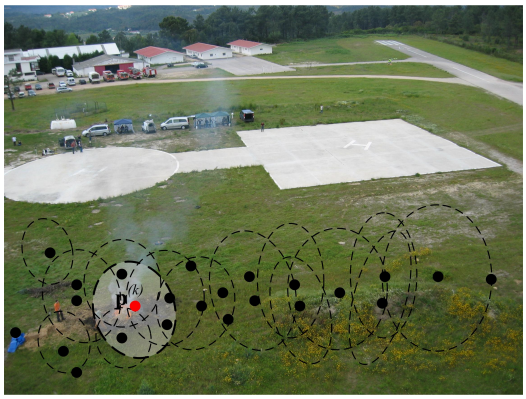

Figure 5.3: Likelihood for the case of the cameras. Due to the uncertainties in the sensor position, each particle (black dots) corresponds to a distribution over the pixels on the image (represented by the ellipse).

measurement equations associated to vision are non-linear.

As a particle filter will be used to determine the position $\mathbf{P}_t$ of an object, in this case each particle $\mathbf{x}_t^{(k)}$ will store a potential value for this position $\mathbf{p}_t^{(k)}$.

5.3.1 Likelihood function for cameras

The derivation of the likelihood function for the cameras is equivalent to the model presented in section 4.2.1. Here, it will be considered that the image processing algorithms of UAV j provide the position on the image plane of the object $\mathbf{m}_{j,t}$:

$$\mathbf{z}_{j,t} = \mathbf{m}_{j,t} \tag{5.28}$$

Provided that the image object is associated to the current object, the likelihood function is given by:

$$p(\mathbf{z}_t|\mathbf{X}_t) = p(\mathbf{m}_{j,t}|\mathbf{P}_t) \tag{5.29}$$

i.e., the position of the object on the image plane will be used to update the belief over the position of the object (represented by the particles). The problem of data association will be considered later.

It is needed to obtain the probability $p(\mathbf{m}_{j,t}|\mathbf{p}_t^{(k)})$ for all the particles $\mathbf{p}_t^{(k)}$, that is, the probability that the measured position on the image plane is $\mathbf{m}_{j,t}$ if the true object position is $\mathbf{p}_t^{(k)}$. Recalling (3.55), the (uncertain) position on the image plane $\mathbf{M}_{j,t}^{(k)}$ corresponding to particle k not only depends on the position $\mathbf{p}_t^{(k)}$, but also on the (uncertain) pose of the sensor $\mathbf{Q}_{j,t}$:

$$\mathbf{M}_{j,t}^{(k)} = \mathbf{f}(\mathbf{p}_t^{(k)}, \mathbf{Q}_{j,t}) + \boldsymbol{\varepsilon}_{j,t} \tag{5.30}$$

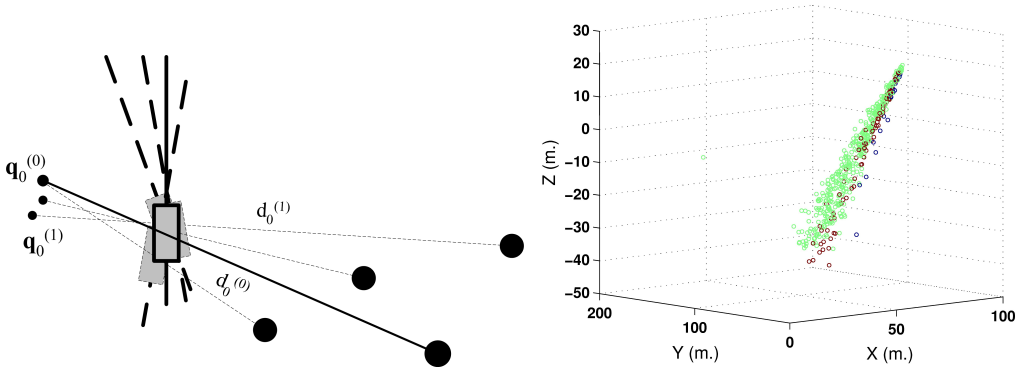

Figure 5.4: Left: scheme of the prior distribution for an object detected on a camera. Different poses for the sensor and potential ranges of the objects are sampled considering the current belief on the sensor position and a uniform distribution for the range. Right: an example of the final particles obtained.

with $\mathbf{f}$ non-linear. The uncertainties in the sensor pose should be explicitly considered when computing the likelihood function. The uncertainties in $\mathbf{Q}_{j,t}$ are propagated into uncertainties on the pixel position by using the UT. Also, an additional source of error $\boldsymbol{\varepsilon}_{j,t}$ on the image plane position can be considered, due to the image processing stage, of zero mean and covariance matrix $\mathbf{S}_{j,t}$. As a result, each particle $\mathbf{p}_t^{(k)}$ corresponds to a Gaussian distribution on the pixel position $\mathcal{N}(\mathbf{m}_{j,t}^{(k)}, \boldsymbol{\Sigma}_{j,t}^{(k)} + \mathbf{S}_{j,t})$ (see Fig. 5.3).

Then, the probability of the actual position of the segmented object $\mathbf{m}_{j,t}$ is given by

$$p(\mathbf{m}_{j,t}|\mathbf{p}_t^{(k)}) = \frac{1}{2\pi\sqrt{|\boldsymbol{\Sigma}_{j,t}^{(k)} + \mathbf{S}_{j,t}|}} \exp(-\frac{1}{2}(\mathbf{m}_{j,t} - \mathbf{m}_{j,t}^{(k)})^T(\boldsymbol{\Sigma}_{j,t}^{(k)} + \mathbf{S}_{j,t})^{-1}(\mathbf{m}_{j,t} - \mathbf{m}_{j,t}^{(k)})) \quad (5.31)$$

5.3.2 Prior distributions

The filter is initiated once a robot of the fleet detects an object on its cameras. The initial set of samples is obtained from the back-projection of the detected object on the image plane. The object must be on the line that contains the camera centre and the image $\mathbf{m}_{j,0}$ of the object. This line can be obtained from the inverse of the equation (3.57).

However, it is not possible to know the range of the object from one image. Therefore, samples are obtained considering a uniform distribution on the depth of the object, between a minimum and a maximum distances d_{min} and d_{max}.

Moreover, the uncertainties on the camera pose $\mathbf{Q}_{j,0}$ are also considered when drawing the samples. These uncertainties are modeled as a multivariate Gaussian, $\mathbf{Q}_{j,0} \sim \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$. The complete procedure for the initial samples is summarized by Algorithm 5.4, and illustrated in Fig. 5.4.

Algorithm 5.4 $(\{\langle \mathbf{p}_0^{(k)}, \omega_0^{(k)} \rangle; k = 1, \ldots, N\}) \leftarrow$ prior_camera$(\mathbf{m}_{j,0}, \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$

1: $X_0 = \{\emptyset\}$
2: **for** $i = 1$ to N **do**
3: sample $\mathbf{q}_0^{(k)} \sim \mathcal{N}(\boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$
4: Determine the direction $\mathbf{v}_0^{(k)}$ from the pose $\mathbf{q}_0^{(k)}$ and $\mathbf{m}_{j,0}$
5: sample $d_0^{(k)} \sim \mathcal{U}(d_{min}, d_{max})$
6: $\mathbf{p}_0^{(k)} = \mathbf{q}_0^{(k)} + d_0^{(k)} \mathbf{v}_0^{(k)}$ /* *In this case, only the position part of* $\mathbf{q}_0^{(k)}$ *is used.* */
7: $\omega_0^{(k)} = 1/N$
8: Add $\langle \mathbf{p}_0^{(k)}, \omega_0^{(k)} \rangle$ to X_0
9: **end for**

5.3.3 Importance function and final filter

As the state considered is static, no prediction equation is considered initially. However, when the effective number of particles is below a threshold (usually 30% of the total number), a resampling step is introduced to maintain the resolution of the filter. After this resample step, the particles are perturbed with additive Gaussian noise $\boldsymbol{\nu}_t \sim \mathcal{N}(0, \mathbf{R}_t)$ in order to explore new zones of the state space,

$$\mathbf{p}_t^{(k)} = \mathbf{p}_{t-1}^{(k)} + \boldsymbol{\nu}_t, \tag{5.32}$$

so that $\mathbf{p}_t^{(k)}$ is sampled from $\mathcal{N}(\mathbf{p}_{t-1}^{(k)}, \mathbf{R}_t)$.

5.3.4 Data association

Finally, the data association problem should be addressed in order to associate the current measurements to the existing tracks; and to associate the incoming information from other robots to the local one.

Scan-to-track association

In the case of camera measurements, it is possible to detect several objects on the image plane, due to clutter or to the presence of a new object. In Gaussian environments, a nearest neighbor approach is usually used for data association, employing the Mahalanobis distance (Section 3.8). For the case of particle representations, there is no direct replacement for this distance. Other features from the images can be used for data association, as some kind of signature (colour, etc). However, if nothing more is known, a Maximum Likelihood approach is used for data association.

If there are several measurements $p = 1, \ldots, L$ provided by UAV j, denote by $\mathcal{H}_p$ the hypothesis that measurement p is associated to the current track while the others are clutter. The probability of obtaining this measure is then:

$$p(\mathbf{z}_{jp,t}|\mathcal{H}_p) = \int p(\mathbf{z}_{jp,t}|\mathcal{H}_p, \mathbf{X}_t) p(\mathbf{X}_t) d\mathbf{X}_t \tag{5.33}$$

The idea is to select the hypothesis $\mathcal{H}_p$ that maximizes the previous likelihood. Given the set of particles, and considering the position on the image plane as the measurement, this probability can be approximated as:

$$p(\mathbf{z}_{jp,t}|\mathcal{H}_j) = \sum_k \omega_{t-1}^{(k)} p(\mathbf{m}_{jp,t}|\mathcal{H}_j, \mathbf{p}_t^{(k)}) \tag{5.34}$$

where the different terms $p(\mathbf{m}_{jp,t}|\mathcal{H}_j, \mathbf{p}_t^{(k)})$ are given by (5.31) for object p. The data association procedure therefore selects the most likely hypothesis.

Track-to-track association

The main objective of this chapter is to show the possibility of implementing decentralized fusion employing particles. Therefore, it will be assumed that the features about the object carry enough information to perform this track to track association, for instance some appearance information. Nevertheless, this is an open problem. In general, other features than position will be needed to perform data association in complex environments.

5.3.5 Communication

In the final scheme, each UAV applies the filter described above locally. At certain time instants, each UAV i sends its particle set $\{\mathbf{x}_{i,t}^{(k)}\}$ to its local neighbors.

As the transmission of particles is demanding, the robots only send them when they have accumulated enough new information. Considering that the last transmission occurred at time $t-k$, this can be qualitatively estimated analyzing the probability $p(\mathbf{z}^{t-k:t}|\mathbf{z}^{t-k})$. This probability can be factorized as:

$$p(\mathbf{z}^{t-k:t}|\mathbf{z}^{t-k}) = p(\mathbf{z}_t|\mathbf{z}^{t-1})p(\mathbf{z}_{t-1}|\mathbf{z}^{t-2})\cdots p(\mathbf{z}_{t-k+1}|\mathbf{z}^{t-k}) \tag{5.35}$$

If the probability of obtaining the new measurements since the previous transmission instant is low, it would indicate that the measurements are informative enough. As $\sum_k \omega_t^{(k)} \approx p(\mathbf{z}_t|\mathbf{z}^{t-1})$, it is possible to calculate the previous product accumulating the non-normalized weights at each time t. When this product drops below a threshold, the likelihood of the measurements is low, and therefore the robot has accumulated new evidence to be transmitted.

5.4 Results

Some simulation runs are used to show the fusion procedure and to compare the two options presented for fusion. Then, the algorithms are employed in an actual experiment of fire detection and localization.

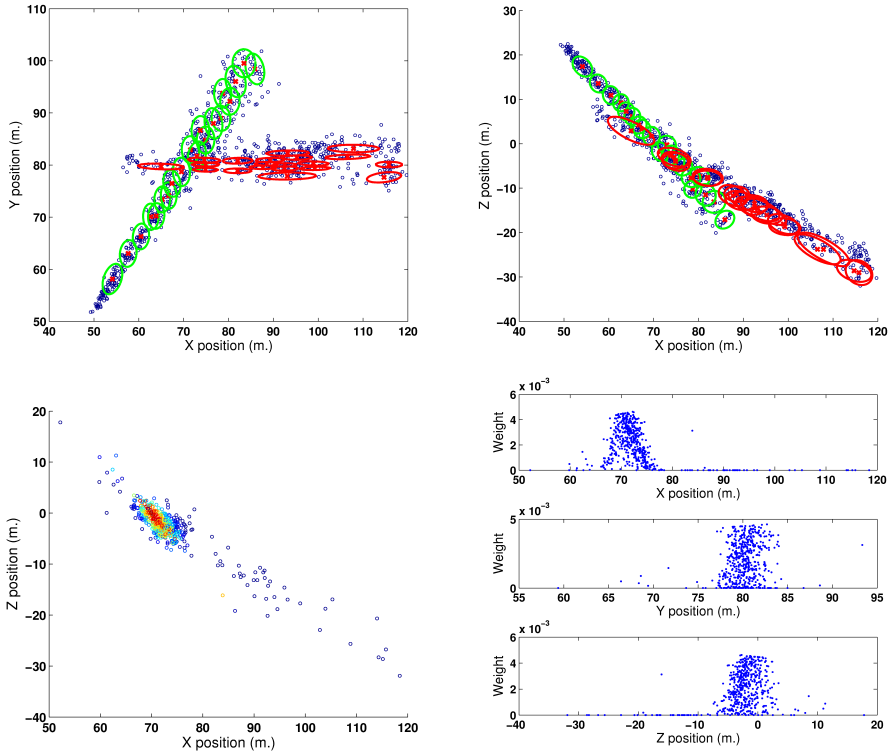

Figure 5.5: Top: The local set of particles of UAV 2 and the received set from UAV 1, together with their GMM approximation. Bottom: particles after fusion

5.4.1 Detection and localization of a static object

The first experiment consists of a simulated scenario in which 2 UAVs collaborate to locate a static object. The two UAVs are commanded to fly over a zone where a potential alarm is located. The two schemes presented above, the one based on a GMM approximation and the LOP-based fusion, are compared.

Figure 5.5 presents an example of the fusion procedure when one of the UAVs receives the particle set from its companion. The GMM approximation is shown, and also the final particle set after updating the weights and resampling to the original number of particles.

Figure 5.6 shows the estimated position of the object for both UAVs for the case of the GMM approximation. It can be seen how receiving information reduces the estimated variance. In this case, the fusion produces a drastic reduction in the variance, as non-compatible hypotheses (particles) receive very low weights (and then are rejected in the resampling step). At the end, the UAVs tend to have the same belief about the position of the object.

Figure 5.7 shows the results employing the linear opinion pool. In this case, the reduction in the estimated variances due to fusion is lower than in the previous case, although the UAVs tend to have the same belief state. This is the main difference between the approaches: in the LOP-based

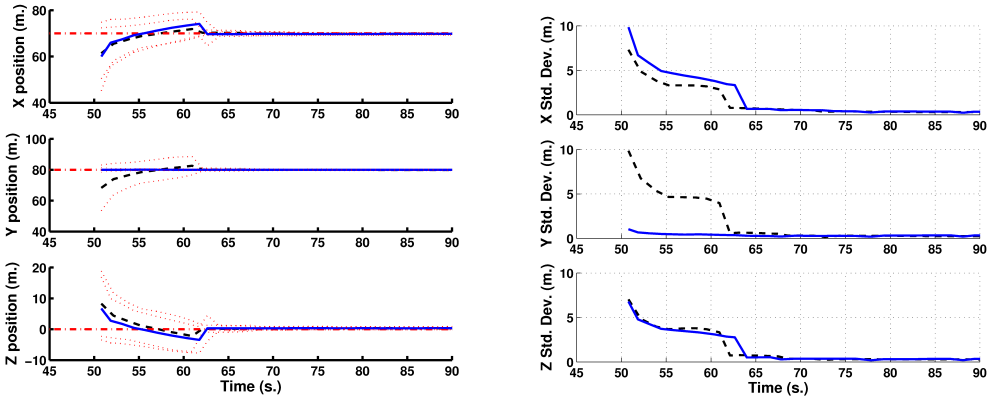

Figure 5.6: Estimated position of the object (left) and estimated standard deviation (right) for both UAVs using the fusion method based on GMMs. The actual alarm position is also shown (dash-dotted).

fusion, each UAV obtains a particle distribution that represents a consensus among the beliefs in the fleet. In the Bayesian case, only the intersection hypothesis is maintained.

The evolution in the particle sets for the GMM case is described in Fig. 5.8. It can be seen how, after fusion, the particles remain in the zone of the state space covered by both sets of particles.

The evolution in the case of the LOP-based fusion is described in Fig. 5.9, where it can be seen the particle set of UAV 2 before, while and after receiving the particle set of UAV1 (in one of the occasions that communication occurs). In this case, the receiving UAV summarizes the belief state of both UAVs within its particle set.

A similar experiment is considered with 3 UAVs. Fig 5.10 shows the trajectories of the UAVs and the particle set for two of them at a given instant. The results of the estimated position for the three UAVs can be seen in Figs. 5.11 and 5.12. Again, all the UAVs reach the same belief, but the GMM approach gives a higher information gain. However, it is slightly overconfident (as in this case, the three UAVs will suffer from rumor propagation) compared to the LOP-based fusion.

Bandwidth and computational complexity

Regarding the bandwidth employed, Fig. 5.13 shows the size of the messages exchanged in this last experiment by UAVs 2 and 3, for the case of 200 particles. Each particle consists of 4 double precision numbers, and then each message is around 6400 bytes (plus the overhead of the communication protocol, which is quite lightweight, and is described in Section 8.3). The transmissions occurs following the scheme of Section 5.3.5. Usually, higher numbers of particles are employed, and therefore the message size required is bigger.

The main drawback of the GMM approximation is the computational complexity. Under a Pentium 1.86 GHz, with 1 GB of RAM memory, which is running the processes of UAV motion, a simple image capture simulator and the decentralized estimator (for each UAV), the GMM approximation

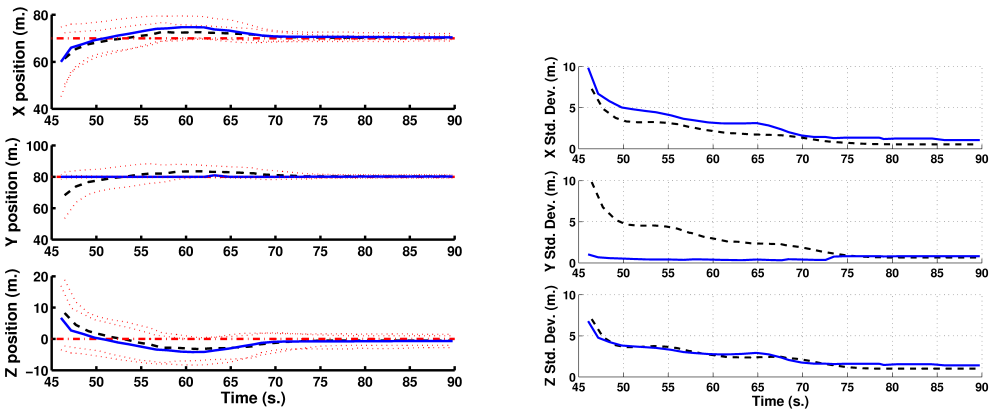

Figure 5.7: Estimated position of the object (left) and estimated standard deviation (right) for both UAVs using the fusion method based on LOPs. The actual alarm position is also shown (dash-dotted).

for 200 particles takes around 1 second, and for 500 particles can last more than 2 seconds. The LOP fusion is, on the other hand, very fast when compared to this method.

5.4.2 Experimental results

The same algorithms have been applied over a set of data gathered during the experiments described in Chapter 8. The experiment aims to the precise localization of a fire alarm from the images gathered by two UAVs

In the results shown here, a flight of one UAV is divided into two different data sets in order to simulate two different UAVs. Figure 5.14 shows two of the images gathered from different points of view, with the particles overlaid.

The two methods described above have been applied in the same experiment. Figure 5.15 shows the results. The alarm is situated approximately at the point $(620, 440)$

It can be seen how the GMM-based method allows reducing more quickly the associated uncertainties. Using both approaches, the UAVs estimate a similar position, although the final standard deviation estimated by the LOP-based fusion is bigger.

Finally, Fig. 5.16 shows the results obtained with the decentralized particle filtering method (employing the LOP fusion) for the fire detection and localization experiment described in Section 3.9.4. The assumptions considered there for the Information Filter approach are here relaxed, as the objects are not assumed to be on the ground. It can be seen how the UAVs are able to estimate the correct position of the fire alarm.

5.5 Discussion

In this chapter, a final belief representation for cooperative perception has been analyzed: particle filters. Particle filters allow considering very general distributions under non-linear prediction and measurement functions. The extension to multi-robot perception is presented. The robots exchange beliefs expressed as particles, and the receiving robot incorporates the companion belief into its own belief.

The chapter has analyzed two potential approaches for fusion employing particles. The first approach is based on the Bayesian framework presented in the first chapters. It actually allows the robot team to fuse and reach a common belief state. The main drawback is that it requires a functional approximation of the belief to perform the fusion operations, which can be costly in terms of computation power. Also, it is not ensured the consistency due to rumor propagation for general topologies.

Another approach, based on consensus theory (the LOP approach) has been also considered. The opinion pools try to reach a consensus among the robots. As a consequence, the information gain is lower, but the fused beliefs are more conservative. They do not require recovering a functional representation of the belief, and therefore they lead to simpler and more efficient fusion rules.

Particle filters present a drawback due to the required bandwidth to transmit the particle set, particularly if the number of particles and the dimension of the state are large. In the GMM-based scheme, a possibility would be to transmit directly the functional approximation instead of the particles, like transmitting the GMMs.

Another potential approach, that could be adequate for some applications, is to use locally the particle filters until the belief state can be well modeled by a Gaussian distribution, and then switch to the decentralized IF of Chapter 3.

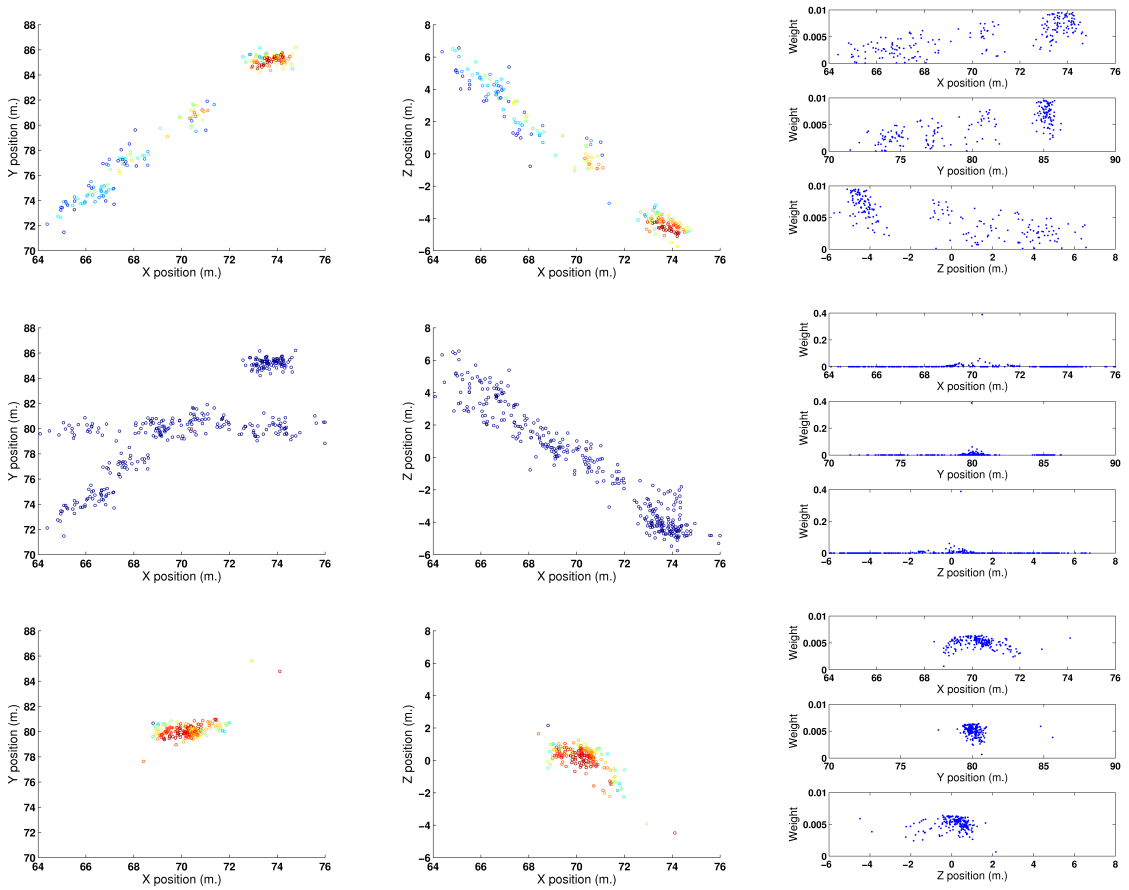

Figure 5.8: Top row: particle set of UAV 2 before receiving information from UAV 1. Middle: during one iteration, the particle set of UAV 1 is included into the local filter, weighting its particles using the GMM approximation of each belief. Bottom row: particles in the next iteration after resampling. The GMM fusion assigns very low weights to particles that are not in the intersection of both beliefs.

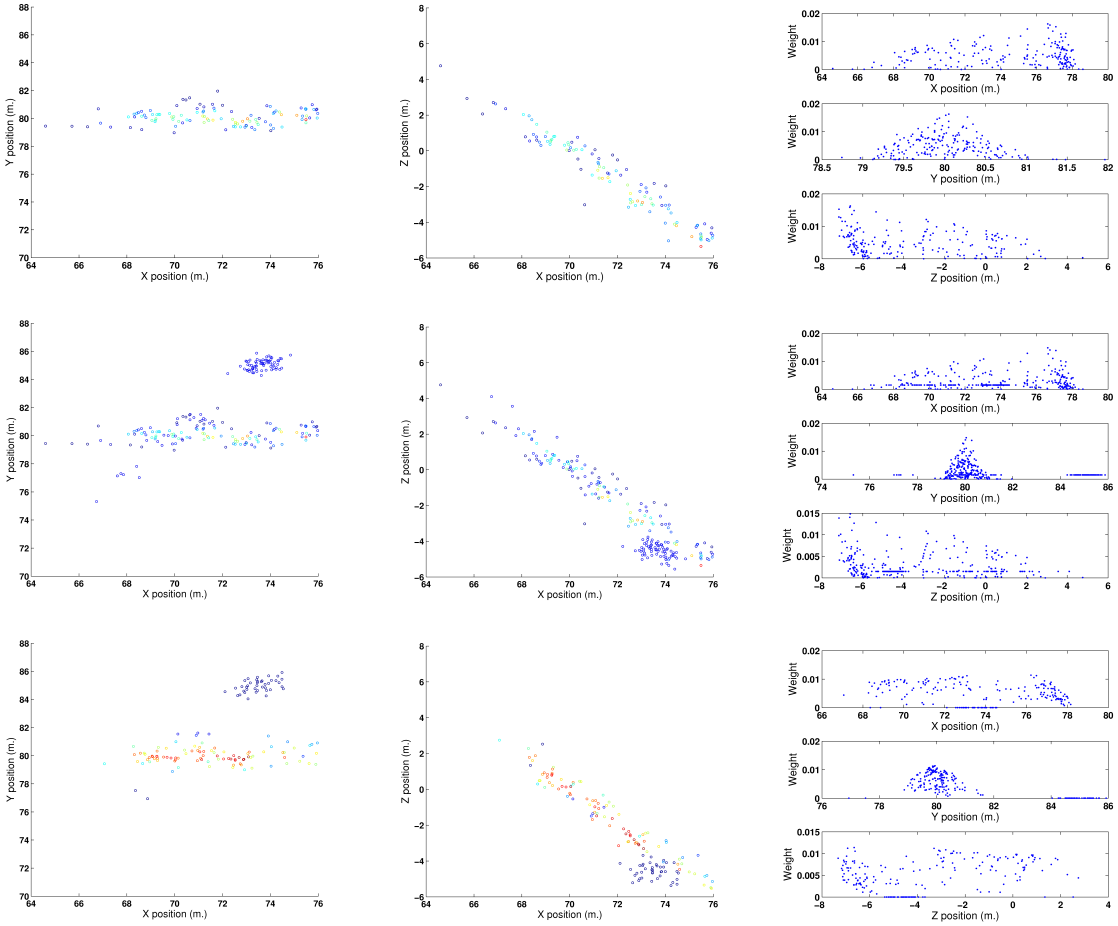

Figure 5.9: Top row: particle set of IAV 2 before receiving information from UAV 1. Middle: during one iteration, the particle set of UAV 1 is included into the local filter, weighting its particles using the LOP fusion procedure. Bottom row: particles in the next iteration after resampling. The LOP fusion maintains a consensus distribution. In general, this distribution will have a higher density of particles around the true value of the state.

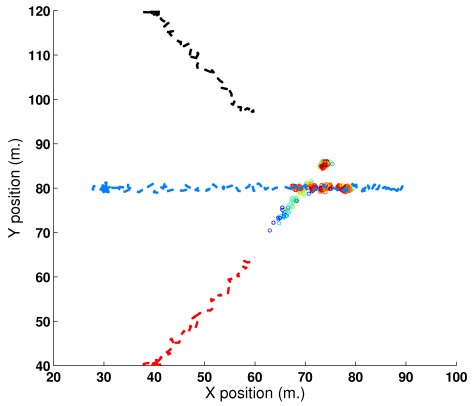

Figure 5.10: Experiment of 3 UAVs localizing one object. Dashed: trajectories of the three UAVs. The circles denote the particle set for two of the UAVs at a given instante. It can be seen how the fleet can benefit from the fusion.

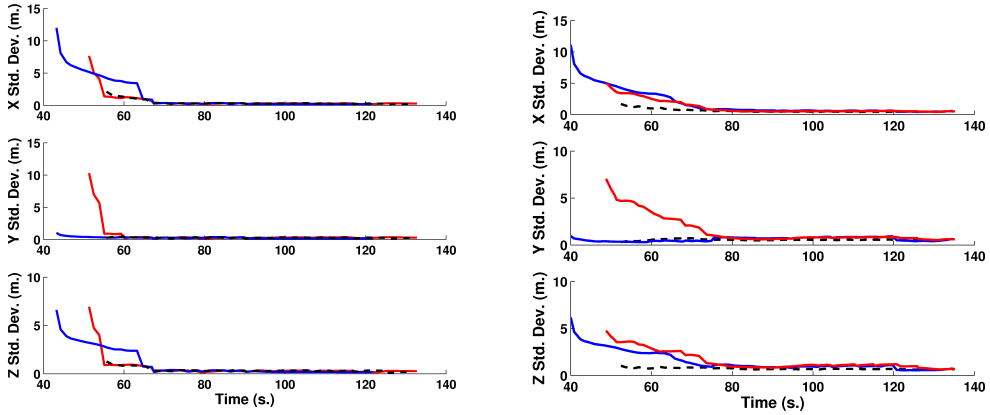

Figure 5.11: Estimated standard deviation obtained by the three UAVs. Left: fusion based on GMM approximation. Right: fusion based on LOP.

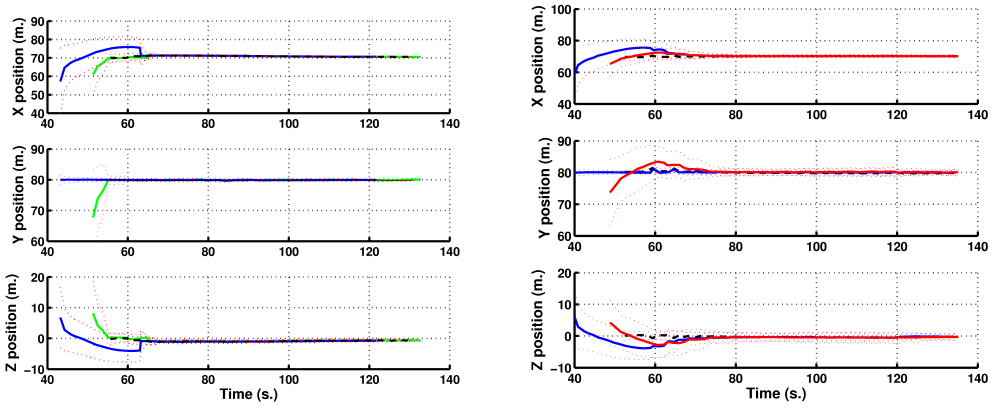

Figure 5.12: Estimated position obtained by the three UAVs. Left: fusion based on GMM approximation. Right: fusion based on LOP. It can be seen how the GMM fusion produce slightly overconfident results.

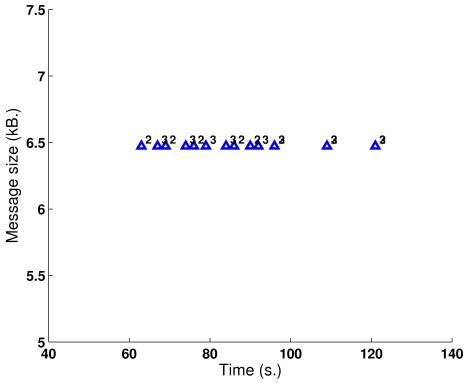

Figure 5.13: Bandwidth in transmission for UAVs 2 and 3 in the experiment.

Figure 5.14: Fire detection experiment 1. Particles projected onto two images gathered by the two UAVs. The different points of view allow reducing the uncertainties.

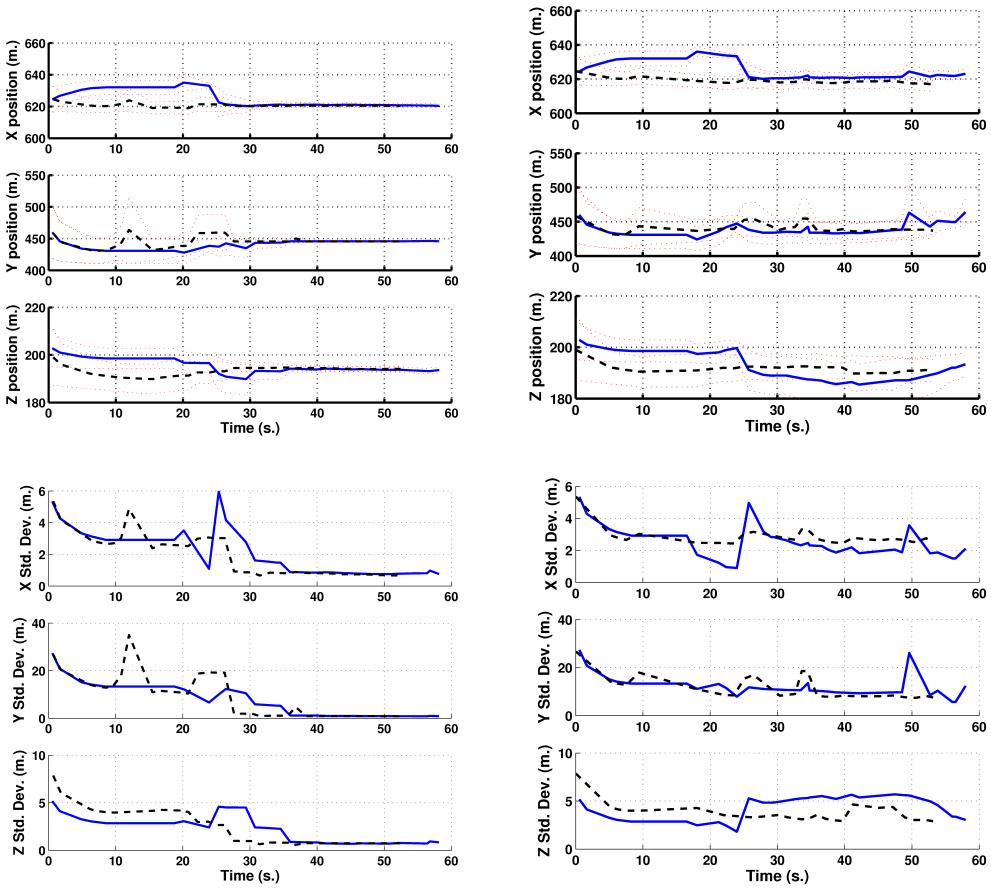

Figure 5.15: Results obtained by the two UAVs for the fire detection experiment 1. Left: fusion based on GMM approximation. Right: fusion based on LOP. Top: estimated position of the alarms. Bottom: estimated standard deviations.

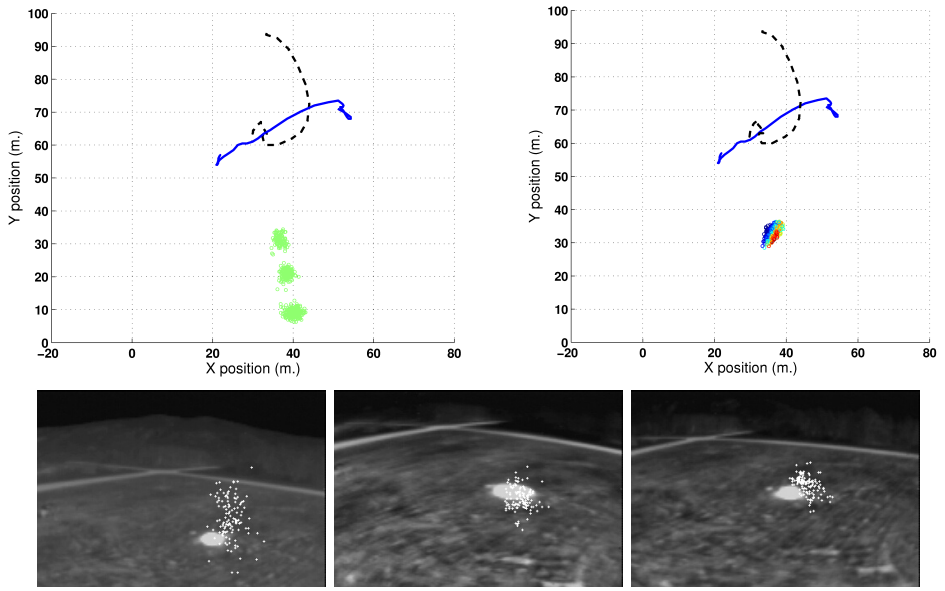

Figure 5.16: Fire detection experiment 2. Top: Particles representing the alarm position obtained by one of the UAVs, at two time instants. The trajectories of both UAVs are also shown. Bottom row: the particles overlaid on the images gathered by the same UAV. It can be seen how the particles agglomerate around the correct position.

Many of those trees were my friends
creatures I had known from nut and
acorn; many had voices of their own
that are lost for ever now.
TREEBEARD, IN J.R.R. TOLKIEN'S
The Lord of the Rings, The Two Towers (1954)

Chapter 6

Cooperative perception algorithms for forest fire monitoring

The previous chapters described techniques for cooperative detection, localization and tracking by means of a fleet of heterogeneous UAVs. These techniques have been tested for fire detection and localization.

Once an event of interest has been declared, confirmed and localized (for instance, a fire), it can be very interesting to obtain detailed views of the event, or even to estimate finer details of it. The high manoeuvrability of UAVs with hovering capabilities, such as helicopters, makes them an ideal tool for this kind of applications. This phase will be called *monitoring* the event. The parameters to be estimated strongly depend on the particular application considered. This chapter presents the potential application of UAVs for fire monitoring.

The chapter begins with a description of the fire monitoring problem, and some current automatic approaches in the literature. Then, the chapter presents algorithms for multi-view fire monitoring using static and aerial cameras on board UAVs. Results from experiments involving controlled forest fires are presented.

6.1 Problem description

Fire monitoring is related to the computation in real-time of the evolution of the most important parameters related to the fire propagation. Figure 6.1 shows the state of a controlled fire at several time instants. Among the most important parameters for fire fighting management are: the shape and position of the fire front, its rate of spread (how this front evolves with time) and the maximum height of the flames (Viegas, 1998; Viegas, 1999). Figure 6.2 presents an schematic description of these parameters.

If available, this information, integrated within a Geographical Information System (GIS), can be used by the fire brigades for fire fighting planning, for instance by predicting the potential evolution of the fire, determining the optimal location of fighting means, etc. Most of monitoring information is provided nowadays by experts that examine visually the evolution of a fire. Hence, the

Figure 6.1: Different stages in the evolution of a fire.

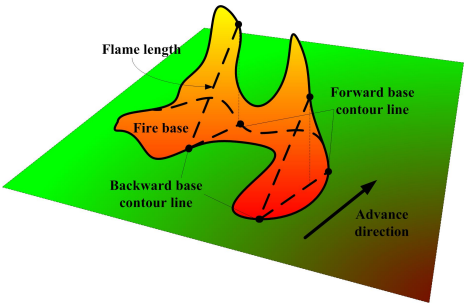

Figure 6.2: Schematic description of the parameters of interest. The system considers as main parameters the position of the fire front and the maximum height of the flames, which indicates the strength of the fire.

development of an automatic system based on computer vision for online fire monitoring is definitely very helpful. The main objective of a system of this kind is, then, the determination, in real-time, of the current position of the fire front in geographical coordinates. Also, the system should provide other interesting parameters of the fire, such as the maximum height of the flames.

6.1.1 Current approaches

Fire fighting usually involves activities before fire, during fire and after fire. Traditionally, information extraction for fire fighting support has been done by experts, directly on the terrain or analyzing data provided by towers, satellites or other means. However, some systems have been developed in order to automatically extract the relevant information from several sources of information.

Before fire activities include the surveillance of areas and the generation of risk maps. Automatic systems, mainly based on satellite imagery and data, have been developed for these tasks (San Miguel Ayanz et al., 2005). Although satellite-based surveillance systems have been proved to be valuable for surveillance in large and homogeneous regions, their main drawback is that the spatial and temporal resolutions of the data they produce are still low for some fire fighting activities.

The main activities *during fire* are *fire detection* and *monitoring*. There are automatic systems for fire detection, although most of them are based on cameras placed on ground systems. For instance, the BOSQUE system, which relies on infrared cameras and automatic image processing techniques for fire detection, including false alarm rejection (Arrue et al., 2000). Colour ground

cameras are also used for autonomous detection of forest fires; and thus, they are considered in the AWISS system (de Vries and Kemp, 1994), or in the ARTIS-FIRE system (Dierre et al., 1999), where smoke-plume segmentation with visual cameras is used for forest fire detection. A system including smoke plume detection and communication facilities was developed in the frame of the AFFIRM project (Den Breejen et al., 1998). Smoke plume detection using image processing is also considered by Gómez-Rodríguez et al. (2002), where wavelet-based optical flow computations are combined with illuminance values for reliable smoke classification.

Other sensors have been also employed for automatic forest fire detection, like LIDAR (Utkin et al., 2002; Utkin et al., 2003), due to its interesting characteristics for smoke plume detection in situations where visual cameras cannot operate (as, for instance, during night). These systems have some drawbacks, such as the coverage and the lack of reliability of the automatic detection in changing environmental conditions.

The use of satellites for fire detection has been also considered (Rauste, 1996; Kelhä et al., 2003). Besides, manned airborne-based systems have been also used, like in the Airborne Wildfire Intelligence System (AWIS), in Canada (Campbell et al., 2002). The previous chapters have shown some results of the use of UAVs for automatic forest fire detection as an application of the techniques described. These results will be complemented with those presented in Chapter 8, in which is considered one of the first demonstrations of a system of this kind.

Fire monitoring is usually performed by experts, that estimate, visually or from images gathered by cameras, the rate of spread and height of the flames. Also, photogrammetric techniques have been applied to images taken of a fire for a posterior analysis of the fire evolution. More recently, airborne systems are used in order to have a broad overview of the fire evolution, but still the monitoring activities are carried out by people. Satellite-based systems have been also proposed for forest fire monitoring (Chuvieco and Martin, 1994; Gonzalo, 1998; Rauste, 1996). As commented above, the temporal and spatial resolutions of these systems are still very low for the requirements of forest-fire fighting in many cases.

There are some fire analysis techniques based on computer vision for fires carried out in laboratories, as (Martínez de Dios et al., 2006a; Pastora et al., 2006). However, the application of the same techniques to non-controlled outdoor environments is not considered in those papers.

After fire activities usually include embers detection and burnt-area mapping. Again, this is usually done by experts onboard aircraft means that fly over previously burnt areas.

The chapter presents the application of aerial robots for the fire monitoring. As commented, manned helicopters or airplanes are often used, but they are expensive and there is a high risk for humans involved in the operation. The use of UAVs can help reducing the risk associated to translate humans on board planes or helicopters to places close to fire.

The idea of using robots for fire fighting was first proposed by Bradshaw (1991), which describes the functional and mechanical design of a ground robot for fire detection and first intervention in indoor environments. Also, Amano et al. (2001) show the design of a climbing robot for helping fire activities in buildings. These robots can be considered as intervention robots in a very hostile environment. However, the practical application of these robots in field forest-fire fighting still

requires significant research and development efforts.

The use of UAVs in forest-fire fighting scenarios has been analyzed in the FiRE project in the United States and in the COMETS project in Europe.

The ALTUS UAV, an evolution of the Predator UAV, has been demonstrated in fire experiments in the FiRE project (Ambrosia, 2002). The data received at the ground station are geo-referenced imagery about the fire. The processing time from data acquisition onboard the UAV, through satellite uplink/downlink, processing into a geo-referenced image data file, to a fire manager in a remote control centre is less than an hour. A 24-hour mission, using a UAV to collect data about selected target fires supplying real-time data by telemetry was planned for 2006 in the framework of the USA WRAP (Wildfire Research and Applications Partnership).

While the FiRE project considers a single and complex UAV on these tasks, the COMETS project addresses the use of a team of simpler UAVs that cooperate in fire detection and monitoring tasks. Very few work has been identified considering multiple UAVs in the task of fire perception. One of them is the work presented in (Casbeer et al., 2005), where the feasibility of the application of a team of small (low-altitude, short endurance) UAVs to cooperatively monitor and track the propagation of large forest fires is explored. The paper provides simulations using a six degree of freedom dynamic model for the UAVs and a numerical propagation model for the forest fire, but results in actual fire fighting activities are still to come. Also, in (Zhou et al., 2005), the authors present a method for orthorectification of images gathered from an UAV, for their application in fire monitoring activities. They discuss particular problems that have to be faced in the case of forest areas, and present very preliminary results on aerial images gathered from a conventional aircraft. However, no actual fire monitoring results are presented.

6.2 A fire monitoring system employing static and aerial cameras

Part of the material described in this chapter has been developed within the SPREAD European project. One of the main objectives of this project was the validation of analytic and qualitative models of fire behavior. For this reason, field experiments of controlled fires were carried out (Viegas et al., 2002). In order to validate those models, fire monitoring data had to be gathered from the experiments. The objective, therefore, was to develop an automatic perception system based on cameras able to determine the parameters described in real-time.

Automatic perception of forest fires is a challenging computer vision problem, with a significant number of sources of errors. To increase the accuracy of the measurements, the perception system designed takes advantage of complementary -and sometimes redundant- information by considering images from multiple cameras. The system exploits two types of complementarities: cameras at different locations in order to obtain images of the phenomenon at different views (aerial, frontal, lateral views), and cameras with different spectral range (visual and infrared). Figure 6.3 shows different examples of the images considered. Moreover, the use of aerial vehicles is considered to gather information about the fire, which can be very helpful in actual fire fighting activities (as in

Figure 6.3: Input data considered. The first two images correspond to different points of view of one experiment. Also, cameras of different modalities are considered. The two last images are a visual and an infrared image of another experiment. Cameras in different positions and/or of different modalities provide complementary information that can be used to increase the robustness of the system.

general it will be difficult to access places where the sensors and cameras can be situated).

6.2.1 Hardware description

The system consists of a central processing station that receives the data and obtains the current parameters of the fire in real-time. As it has been said, the system considers as input data mainly images, both infrared and visual, gathered from static and aerial cameras on board robots, although other sensors are considered as well. A short description of the hardware employed is presented below.

Infrared cameras

The infrared cameras preferred in the proposed system have spectrum response in the mid-infrared band $[3-5]\mu m$, which according to Wien's Law (Hudson, 1969) has optimal sensitivity in the range of temperature of forest fires. Besides, mid-infrared is the band with the lowest atmospheric absorption, (Hudson, 1969). In the mid-infrared band the flames have relatively high transparency (Den Breejen et al., 1998). Thus, it is not possible to obtain accurate flames estimations but, on the other hand, the observations of the fire obtained are not perturbed by smoke (this is very important for a reliable perception of the fire). Infrared images are used to provide accurate estimations of the fire base such as its location. Figure 6.3, right, shows an infrared image of this kind. Although better performance is obtained with cameras in the mid-infrared band, cameras in the far-infrared band, $[7-14]\mu m$ have been applied as well.

Visual cameras

Besides infrared cameras, common off-the-shelf visual cameras are considered. They can provide measures of the flames such as the flames height and also measures of the fire base such as location and shape of the fire-front. The cameras are equipped with optics for close-up views of the fire. Although visual cameras have lower costs, sometimes they cannot provide accurate measures, particularly when smoke occludes the fire front (see Fig. 6.3, third image).

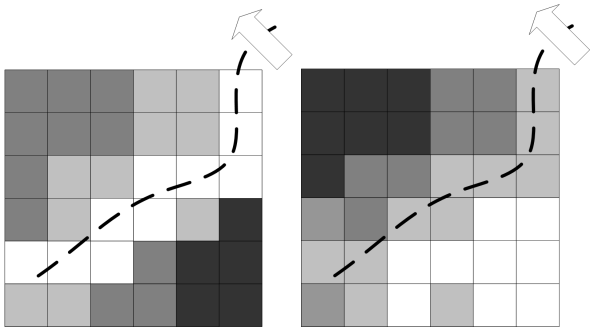

Figure 6.4: The state to be estimated is composed by the the evolution of the fire front and the burnt zone. An evidence grid is employed to store the current belief. Each cell stores two probability values referring to the fire front position and the extension of the burnt zone.

Other sensors

The location of the cameras must be computed in order to anchor the image information in geographical coordinates. The sensors on board the UAVs are used to estimate the pose of their cameras. The system considers also as inputs information provided by sensors like GPS receivers, maps and information about known landmarks for the estimation of the pose of the static cameras. Section 6.8 will show examples of the actual deployment of the system during forest fire experiments.

Moreover, video transmitters are used for receiving in the processing station images from distant points of views.

6.3 Algorithm description

The first question is to devise a convenient representation of the information related to the fire. During the fire detection phase, the fire can be approximated as an object with a definite position, as it was made in the previous chapters. However, after ignition, a fire will propagate, changing in size, shape, etc.

The representation chosen here for autonomous perception consists of an evidence grid. The scenario considered is divided into a rectangular grid, in which each cell stores two probability values $\{f_k, q_k\}$, in the manner of an occupancy grid as described in Chapter 4 (see Fig. 6.4). These two values correspond to the probability of being fire at each cell, $p(F_{k,t} = 1)$, and the probability that the cell is burnt $p(Q_{k,t} = 1)$ (that is, that all the fuel in each cell is completely wasted).

The system should obtain online a belief about the current state of the fire evolution. The state to be estimated consists on the particular status of all the cells. As in Chapter 4, an initial approximation will be done, and the belief state will be approximated by the marginals of each cell in the grid.

$$bel(\mathbf{X}_t) = p(F_t, Q_t|\mathbf{z}^t) \approx \prod_k p(F_{k,t}, Q_{k,t}|\mathbf{z}^t) \tag{6.1}$$

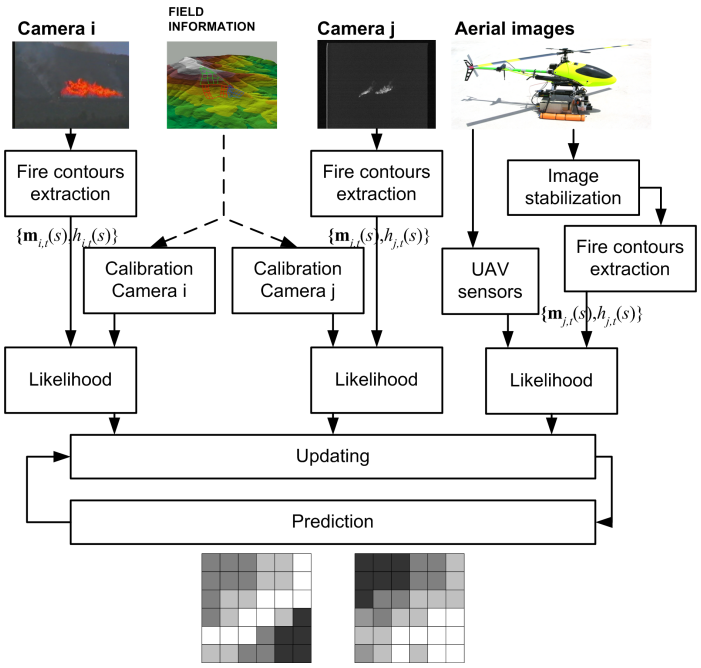

Figure 6.5: The state of the grids is estimated incorporating data obtained from the set of cameras. The prediction step incorporates the increase in uncertainty due to the motion of the fire.

Moreover, a separated filter will be used to estimate $p(F_{k,t}|\mathbf{z}^t)$ and $p(Q_{k,t}|\mathbf{z}^t)$.

The integration of new information provided by the cameras is done by using a discrete Bayes Filter for each cell of the grid, similar to that presented in Chapter 4. The steps involved in the estimation are summarized in Fig. 6.5.

Besides, the system maintains an estimation of the maximum height of the flames in each time instant H_t, as a Gaussian variable. A Kalman filter is employed to update this estimation with the data provided by the different cameras.

6.3.1 Prediction model

As shown in Fig. 6.5, the system allows considering a prediction step in the estimation, so that fire propagation models could be included. There are many aspects that influence the propagation of fire, like the slope and solar exposition of the terrain, the moisture content of the vegetation, the meteorological conditions -wind, air humidity-, etc (Viegas, 1998). It is not the objective of this thesis to deal with these aspects.

The prediction model depicted here is very simple, and considers two different relations: a temporal relation and a spatial relation among cells. Its main objective is, in one hand, to incorporate a kind of memory in the estimation process, so that the fire does not propagate "backwards" through zones previously visited (this is the role of Q). Also, a spatial prediction is performed in order to smooth the estimated evolution of the fire fronts.

Although there is a separated filter for each variable, both are not independent, and therefore the dependence should be marginalized out during the prediction phase. The transition probability $p(\mathbf{X}_t|\mathbf{X}_{t-1})$ for each cell can be decomposed as:

$$p(F_{k,t}|F_{k,t-1}) = \sum_{Q_{k,t-1}} p(F_{k,t}|F_{k,t-1}, Q_{k,t-1})p(Q_{k,t-1}) \tag{6.2}$$

$$p(Q_{k,t}|Q_{k,t-1}) = \sum_{F_{k,t}} p(Q_{k,t}|F_{k,t}, Q_{k,t-1})p(F_{k,t}) \tag{6.3}$$

The different terms are characterized by one main parameter, β, which is the probability that a cell is completely burnt if there was fire since the previous time instant:

$$p(Q_{k,t} = 1|F_{k,t}, Q_{k,t-1}) = \begin{cases} 1 & \text{if } Q_{k,t-1} = 1 \\ 0 & \text{if } Q_{k,t-1} = 0 \text{ and } F_{k,t} = 0 \\ \beta & \text{if } Q_{k,t-1} = 0 \text{ and } F_{k,t} = 1 \end{cases} \tag{6.4}$$

On the other hand, the temporal evolution of the fire probability is given by:

$$p(F_{k,t} = 1|F_{k,t-1}, Q_{k,t-1}) = \begin{cases} 0 & \text{if } Q_{k,t-1} = 1, \forall F_{k,t-1} \\ 0 & \text{if } F_{k,t-1} = 0 \, \forall Q_{k,t-1} \\ 1 & \text{if } Q_{k,t-1} = 0 \text{ and } F_{k,t-1} = 1 \end{cases} \tag{6.5}$$

that is, if the cell is completely wasted, there cannot be fire in that cell.

The spatial relation should encode the effect due to fire propagation. In the simple model considered here, the fire can propagate from one cell to its neighbors and, therefore, the state of its neighbor cells $R(k)$ affects the current state of each cell k. This relation is modeled mainly by parameter ω:

$$p(F_{k,t} = 1|Q_{k,t-1}, F_{q,t-1}) = \begin{cases} 0 & \text{if } q \notin R(k) \\ \omega_q & \text{if } F_{q,t-1} = 1 \text{ and } q \in R(k) \text{ and } Q_{k,t-1} = 0 \\ 0 & \text{if } F_{q,t-1} = 0 \text{ or } Q_{k,t-1} = 1 \end{cases} \tag{6.6}$$

Of course, fire propagation follows more complex laws (it can be transported by wind, if there are trees then the propagation can be through the crowns, etc.). Again, it should be stressed that the main motivation of the motion model is to take into account in the estimation process the uncertainties in the fire front position due to the motion of the fire. It is not the objective to model this propagation. Nevertheless, more complex models of fire propagation could be included within the system.

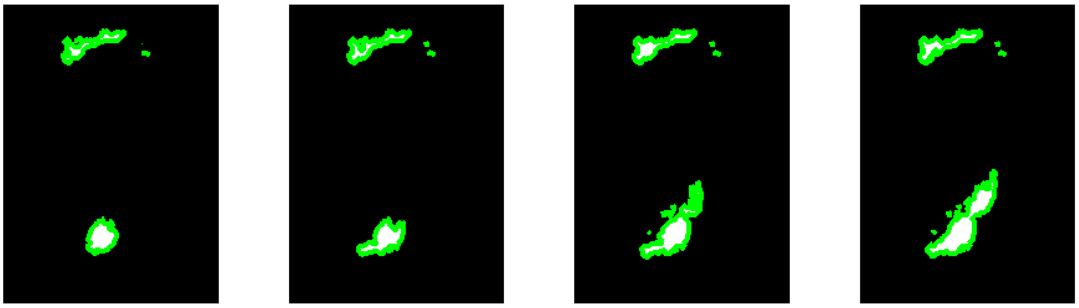

Figure 6.6: The fire front shape can be obtained from the boundaries of the burnt zone. The figure shows the evolution of the burnt zone during one experiment.

6.3.2 Prior belief state and fire front shape computation

Usually, the grid is initiated setting the Bernouilli probabilities $\{f_k, q_k\}$ to a small value for all cells of the grid. Nevertheless, it is straightforward to include prior knowledge into the grid for estimation. For instance, if monitoring is initiated after a detection mission, the f_k value can be set to one for the cells corresponding to the initial estimated position of the fire. Also, knowledge about firewalls can be included into the q_k values of the grid, setting the burnt value to one for places that cannot be crossed by the fire.

On the other hand, it is important to know the estimated position of the fire front. This fire front should be on the boundaries of the burnt zone. Also, the position of the fire front $\{\mathbf{p}_t(s)\}$ should be coherent with cells that maximize the posterior probability of fire.

Therefore, the procedure determines the boundaries of the burnt zone as the contour of regions of cells with burnt probabilities over a given threshold (see Fig. 6.6). These boundaries can be obtained with classical tools for thresholding. From these positions, the final fire front is obtained by considering connected cells on the boundary with high fire probability $p(F_k|\mathbf{z}^t)$.

6.4 Features extraction from images

The information about the fire front position and the height of the flames is encoded in the images as the contour of fire regions on the image plane, so the images gathered from the different cameras are processed in order to obtain those contours.

First, a pre-processing stage is applied, which consists of the following steps:

- A fire segmentation algorithm, that is applied over the images gathered by the cameras. The algorithms employed for fire segmentation in infrared and visual images are described in Appendix B. These algorithms provide as a result a binary image with the fire segmented from the background.

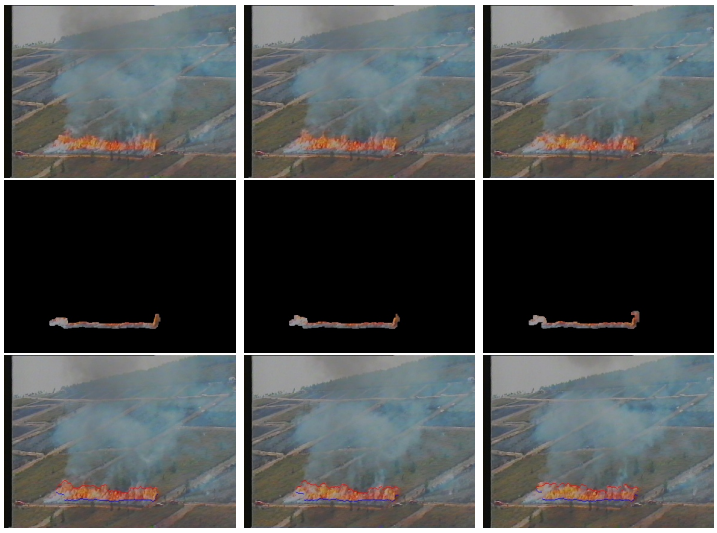

Figure 6.7: Feature extraction. Top row: input images. Middle row: regions obtained by temporal low-pass filtering of the segmented images. Low row: contours obtained.

- After fire segmentation, some binary filters are applied over the segmented images (Haralick and Shapiro, 1992):

  Spurious pixels removing A binary correction algorithm is applied to filter out spurious effects mainly originated by isolated fire and background pixels.

  Region growing algorithm A region growing algorithm is carried out over the segmented images.

  Filling gaps A filling gap operation over the segmented image is carried out to avoid false objects originated by gaps within the fire blobs.

- After this step, the contour of the segmented regions is then obtained.

However, this contour is further characterized in order to distinguish the pixels of the contour related to the fire front and the pixels related to the top of the flames, therefore obtaining the height of the flames in pixel coordinates. The dynamic properties of the fire base and the flames are used for this characterization.

A motion analysis is performed to identify the flame pixels, as the position of the fire-base pixels on the image plane generally change more slowly than the position of the flame pixels (as the flames flicker). The application of a temporal low-pass filter over a sequence of consecutive segmented images is therefore used to filter out the flame pixels. This can be implemented very efficiently as an AND ($\wedge$) operation over the set of binary images (assuming that we represent as `true` the pixels representing fire). If I_t^b is the binary image at time t, fire base pixels will belong to the region determined by the image:

Figure 6.8: A visual (left) and an infrared (right) images of the same fire at approximately the same time instant. Infrared images are not affected by smoke, and therefore they also provided negative information. On the other hand, although the fire front can be occluded by smoke in visual images, they provide accurate measurements of the height of the flames.

$$I_t^{fb} = I_t^b \wedge I_{t-1}^b \wedge \cdots \wedge I_{t-N}^b \tag{6.7}$$

where N depends on the frame rate, and is selected so that N images cover a period of around 1 second, much lower than the usual evolution of the fire base. The pixels in I_t^{fb} have not changed over the last N frames and are likely to be fire-base pixels.

Figure 6.7 illustrates the extraction of fire measures from three consecutive visual frontal images. The original images are shown at the top row. The fire-base pixels resulting from the low-pass filtering of eq. (6.7) are shown at the center. The bottom row shows the original images with the obtained contours overlayed.

Therefore, as a result of the feature extraction algorithms, the measurements $\mathbf{z}_{j,t}$ provided by each camera j are the $M_{j,t}$ pixels corresponding to the fire front and the height of the flames for each point of the front in pixels $\mathbf{z}_{j,t} = \{\mathbf{z}_{j,t}(s), s = 1, \ldots, M_{j,t}\}$. Moreover, it will be assumed that the height $h_{j,t}(s)$ of the flames in meters can be obtained, from the height in pixels, by some means (that will be explained below). Then:

$$\mathbf{z}_{j,t}(s) = \begin{pmatrix} u_{j,t}(s) \\ v_{j,t}(s) \\ h_{j,t}(s) \end{pmatrix} = \begin{pmatrix} \mathbf{m}_{j,t}(s) \\ h_{j,t}(s) \end{pmatrix}, s = 1, \ldots, M_{j,t} \tag{6.8}$$

6.5 Likelihood functions

The measurements obtained with the cameras are used to update the probabilities of each cell of the grid. Also, the heights are used to update the current estimation of the maximum height of the flames. This requires the determination of the measurement model or likelihood function $p(\mathbf{z}_t|\mathbf{X}_t)$.

The measurements are the extracted fire fronts on the images. However, the likelihood of these fire contours is considered differently depending on its origin. For instance, infrared images are not

affected by smoke, and therefore it is possible to employ negative information (that is, if nothing is obtained on the image plane it is very likely that there is no fire), which is not the case of visual images (see Fig. 6.8).

6.5.1 Likelihood for infrared-based measurements

The likelihood functions $p(\mathbf{z}_{j,t}|F_k)$ and $p(\mathbf{z}_{j,t}|Q_k)$ for all cells k should be characterized. Each cell k has an associated 3D position that corresponds to pixel $\mathbf{m}_{kj,t}$ on the image plane of camera j, and that can be obtained for a calibrated and localized camera using the pin-hole model (3.57).

Following the same argument as in Section 4.2.1, the estimated position on the image plane of point $\mathbf{r}_k$ will be affected by the errors in the pose and calibration of the camera. These errors on the estimated position $\mathbf{m}_{kj,t}$ are approximated by a Gaussian distribution of zero mean and a certain covariance matrix $\mathbf{\Sigma}_{kj,t}$, which is estimated by using the UT. These errors define a region $\omega(\mathbf{m}_{kj})$ on the image plane given by the pixels within a certain Mahalanobis distance:

$$\omega(\mathbf{m}_{kj}) = \mathbf{m}_j \mid [\mathbf{m}_j - \mathbf{m}_{kj,t}]^T \mathbf{\Sigma}_{kj,t}^{-1} [\mathbf{m}_j - \mathbf{m}_{kj,t}] < th \tag{6.9}$$

If the estimated fire front contour $\mathbf{m}_{j,t}(s)$ passes within this region, the probabilities for cell k are updated accordingly:

$$p(\mathbf{z}_{j,t}|F_k = 1) = \begin{cases} P_D & \text{if } \mathbf{m}_{j,t}(s) \in \omega(\mathbf{m}_{kj}) \text{ for some } s \\ P_F & \text{if } \mathbf{m}_{j,t}(s) \notin \omega(\mathbf{m}_{kj}) \text{ for any } s \end{cases} \tag{6.10}$$

In the case of $p(\mathbf{z}_{j,t}|F_k = 0)$:

$$p(\mathbf{z}_{j,t}|F_k = 0) = \begin{cases} 1 - P_D & \text{if } \mathbf{m}_{j,t}(s) \in \omega(\mathbf{m}_{kj}) \text{ for some } s \\ 1 - P_F & \text{if } \mathbf{m}_{j,t}(s) \notin \omega(\mathbf{m}_{kj}) \text{ for any } s \end{cases} \tag{6.11}$$

In the same way, the probability of being burnt is decreased for the cells corresponding to the fire front. The values P_D and P_F represent the capabilities of the infrared images for the fire front contour detection. By considering P_F it is possible to use negative information in the estimation process. Infrared images provide information about places where there is no fire, as they are not occluded by smoke.

Infrared images are not used, however, for the computation of the maximum height of the flames, as they do not provide accurate measures of the full length of the flames (due to the radiation pattern of the flames).

6.5.2 Likelihood for the visual-based measurements

The likelihood functions relative to the state of the grid $p(\mathbf{z}_{j,t}|F_k)$ are the same as for the infrared camera. However, as the fire front can be occluded by the flames, no negative information is used in the case of visual images. This is done setting $p(\mathbf{z}_{j,t}|F_k) = 0.5$ wherever a pixel is not classified as fire, for any value of F_k.

Visual images are also used to update the estimated maximum height. As it will be seen, under certain circumstances it is possible to estimate the heights of all the points of the contour. Assuming a additive noise model, the measured maximum height is related to the estimated maximum height following:

$$h_{j,t} = H_t + \varepsilon_t \tag{6.12}$$

Again, ε_t is approximated by a Gaussian distribution of zero mean and variance $\sigma^2_{j,t}$. Then,

$$p(\mathbf{z}_{j,t}|H_t) = \eta e^{-\frac{1}{2\boldsymbol{\sigma}^2_{j,t}(s^*)}[h_{j,t}(s^*)-H_t]^2} \tag{6.13}$$

where s^* is the index corresponding to the highest flame.

$$s^* = \arg\max_s h_{j,t}(s) \tag{6.14}$$

6.6 Obtaining metric measurements from the static cameras

The likelihood functions require to know the relation between pixel and world coordinates. This implies that the position and orientation of each camera should be known, and also that each camera must be calibrated. All these data should be computed for the set of static cameras deployed in the field. In operational conditions, it is interesting to simplify the operations needed to obtain this measurement model. Thus, an approximation is made which allows for a simpler method to determine the parameters of the likelihood functions. For the aerial robots, the sensors on board will be used to measure the camera pose in a world frame, therefore reducing the assumptions.

6.6.1 Projective geometry for planar scenes

Consider that the points of the 3D world are on a planar surface. Also, without loss of generality, consider that that plane corresponds to the plane $Z = 0$. If points and pixels are expressed in homogeneous coordinates, following (3.57) the projection of a point $\mathbf{p}_t(s) = \begin{pmatrix} x_t(s) & y_t(s) & 0 \end{pmatrix}^T$ into camera i becomes:

$$\alpha \begin{pmatrix} \mathbf{m}_{j,t}(s) \\ 1 \end{pmatrix} = \mathbf{A}_j \begin{pmatrix} \mathbf{R}_j & -\mathbf{t}_j \end{pmatrix} \begin{pmatrix} x_t(s) \\ y_t(s) \\ 0 \\ 1 \end{pmatrix} = \underbrace{\mathbf{A}_j \begin{pmatrix} \mathbf{r}_{j,1} & \mathbf{r}_{j,2} & -\mathbf{t}_j \end{pmatrix}}_{\mathbf{H}_j} \begin{pmatrix} x_t(s) \\ y_t(s) \\ 1 \end{pmatrix} \tag{6.15}$$

where $\mathbf{r}_{j,1}$ and $\mathbf{r}_{j,2}$ are the first and second columns of the rotation matrix $\mathbf{R}_j$ that refers camera j to the world coordinate system. As in the general pin-hole model, the relation between the real world coordinates $\begin{pmatrix} x_t(s) & y_t(s) \end{pmatrix}^T$ and the corresponding pixel coordinates $\mathbf{m}_t(s)$ is linear considering homogeneous coordinates. The relation is represented by the 3×3 matrix $\mathbf{H}_j$, which is called a *planar homography* (Hartley and Zisserman, 2004). Planar homographies play an important role

here and in other parts of this thesis. Appendix C shows several properties and issues related to them. There are two interesting properties that should be pointed out.

First of all, if the camera centre $\mathbf{t}_j$ is not on the plane, the matrix $\mathbf{H}_j$ is of rank 3 and therefore, invertible. That is, if $\mathbf{H}_j$ is known, it is possible to determine the pixel position in camera j of any point in the plane; and with its inverse it is possible to determine the world coordinates of a point on the plane given its pixel position at camera j (that is, to geolocate the pixel).

$$\alpha' \begin{pmatrix} \mathbf{p}_t(s) \\ 1 \end{pmatrix} = \mathbf{H}_j^{-1} \begin{pmatrix} \mathbf{m}_{j,t}(s) \\ 1 \end{pmatrix} \tag{6.16}$$

The second property comes in the computation of $\mathbf{H}_j$ (or its inverse). For the operational functioning of the system, the projection model should be obtained from measurements obtained directly with the cameras (usually, matches between points on the image plane and points in the world frame whose coordinates are known). The computation of the general projection matrix $\mathbf{A}_j \begin{pmatrix} \mathbf{R}_j & -\mathbf{t}_j \end{pmatrix}$ from image measurements, using for example the algorithms of Tsai (Tsai, 1987) or Zhang (Zhang, 2000), requires a considerable number of correspondences between pixels on the image plane and points in 3D real-world coordinates. It is not realistic to know the location of many points in the forest scenes.

However, an homography is defined up to an scale factor, so it only has 8 degrees of freedom. Knowing four correspondences between points on the image plane and points on the terrain, the homography $\mathbf{H}_j$ can be computed. Moreover, in this case there is no need to compute explicitly the internal calibration of the camera $\mathbf{A}_j$. If available, more than 4 points (8 to 10 in the normal case) are chosen to increase the accuracy. Appendix C.1 shows an algorithm for the homography computation from point matches. Moreover, it is posible to obtain an estimation of the errors on the computed homography $\mathbf{H}_j$ (see Section C.2), which will be considered in the estimation process.

Thus, once $\mathbf{H}_j$ is known, it is possible to relate points on the terrain (the cells of the grid) with pixels on each camera plane. Moreover, as the homography is invertible, it could be also possible to determine the world coordinates corresponding to pixels on the image plane.

6.6.2 Height estimation

There is also an interesting property in the case of planar scenes, which, under certain circumstances, permits to estimate heights over the reference plane without explicit calibration of the cameras. This can be used to estimate the heights of the flames.

Some definitions are needed beforehand[1]. A camera can be seen as a projective device. Under perspective projection, any given set of parallel lines are imaged as converging lines. Their image intersection is called the *vanishing point* of these parallel lines. It can be seen that the vanishing point $\mathbf{v}$ only depends on the direction of the parallel lines. Geometrically, this point is obtained by intersecting the image plane with a ray parallel to these lines and passing through the camera center $\mathbf{t}$ (see Fig. 6.9, left).

[1] Only in this section, capital letters will be used to represent points in the 3D world, and lower case letters are employed to represent image coordinates.

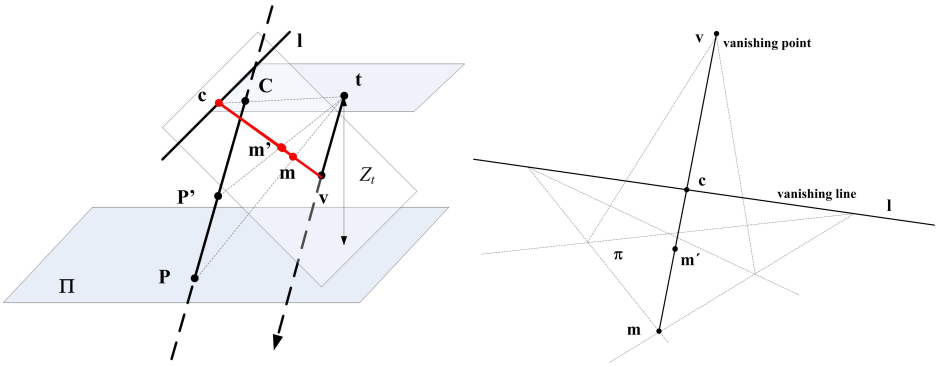

Figure 6.9: Left: Basic geometry for the projection of planes and lines. $\mathbf{v}$, vanishing point of the directions parallel to the segment $\mathbf{P} - \mathbf{P}'$; $\mathbf{l}$, vanishing line of plane Π. Right: the same entities expressed on the image plane.

Similarly, the images of a set of parallel planes in 3D space intersect in a common line called the *vanishing line*. Again, this line $\mathbf{l}$ only depends on the orientation of the plane Π (not on its position), and can be constructed geometrically by intersecting the image plane with one plane parallel to the plane considered and containing the camera centre (see Fig. 6.9, left). It is easy to check that the vanishing points of lines parallel to a plane are located on the vanishing line of that plane. A more detailed description of these concepts can be found for instance in (Criminisi et al., 2000; Hartley and Zisserman, 2004).

Criminisi et al. (2000) showed that, if the vanishing line $\mathbf{l}$ of a reference plane (the terrain in this case) and the vanishing point $\mathbf{v}$ of a reference direction not parallel to that plane are known, then distances from the reference plane along the reference direction can be computed from their imaged points up to an scale factor that can be recovered if one reference length is known.

The geometric scheme is shown in Fig. 6.9. $\mathbf{m}$ and $\mathbf{m}'$ are the image points of the extremes $\mathbf{P}$ and $\mathbf{P}'$ of a segment parallel to the reference direction, whose length $Z = \|\mathbf{P} - \mathbf{P}'\|$ has to be measured. The point $\mathbf{P}$ is on the reference plane Π. $\mathbf{v}$ is the vanishing point of the lines parallel to $\mathbf{P} - \mathbf{P}'$ and $\mathbf{l}$ is the vanishing line of plane Π.

The four points $\mathbf{m}$, $\mathbf{m}'$, $\mathbf{c}$ and $\mathbf{v}$, and their corresponding points $\mathbf{P}$, $\mathbf{P}'$, $\mathbf{C}$ and $\mathbf{V}$ define a cross-ratio (Semple and Kneebone, 1952), which is a projective invariant (and thus, is invariante under perspective projection). Therefore,

$$\frac{d(\mathbf{m},\mathbf{c})d(\mathbf{m}',\mathbf{v})}{d(\mathbf{m}',\mathbf{c})d(\mathbf{m},\mathbf{v})} = \frac{d(\mathbf{P},\mathbf{C})d(\mathbf{P}',\mathbf{V})}{d(\mathbf{P}',\mathbf{C})d(\mathbf{P},\mathbf{V})} \tag{6.17}$$

$\mathbf{V}$ is the backprojection of the vanishing point, and therefore is a point at infinity. Thus, $\frac{d(\mathbf{P}',\mathbf{V})}{d(\mathbf{P},\mathbf{V})} = 1$. $\mathbf{C}$ is a point at the same distance Z_t of the plane Π as the camera centre, as is the backprojection of a point on the vanishing line of plane Π. Therefore, $d(\mathbf{P}',\mathbf{C}) = Z_t - Z$ and the following relation holds:

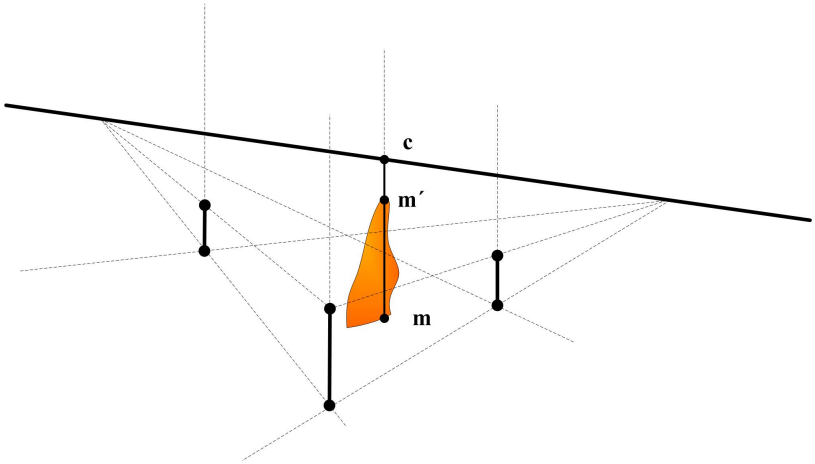

Figure 6.10: The procedure described can be used to compute the height of the flames in meters from its length in pixels $\|\mathbf{m}' - \mathbf{m}\|$, if one reference length is known.

$$\frac{Z}{Z_t} = 1 - \frac{d(\mathbf{m}', \mathbf{c})d(\mathbf{m}, \mathbf{v})}{d(\mathbf{m}, \mathbf{c})d(\mathbf{m}', \mathbf{v})} \tag{6.18}$$

where Z_t is the distance from the camera to the reference plane. In general, this distance cannot be measured directly. However, it can be computed using the same relation (6.18) if a segment with a known length Z_0 is seen on the image plane.

Although mechanisms to compute the vanishing line or the vanishing point are well established, (see for instance (McLean, 1995)), computing the vanishing line of the plane and the vanishing point of a direction are difficult tasks in unstructured environments, such as considered here. A further assumption can be made. The interest is to compute the height of the flames over the terrain, so the reference direction will be the vertical axis over the terrain. If the camera is not tilted, then the camera plane is nearly parallel to this direction, and thus, the vanishing point $\mathbf{v}$ goes to infinity on the image plane (see Fig. 6.10). Then $\frac{d(\mathbf{m},\mathbf{v})}{d(\mathbf{m}',\mathbf{v})} = 1$, and the length of a segment parallel to the reference direction is computed as:

$$Z = \left[1 - \frac{d(\mathbf{m}', \mathbf{c})}{d(\mathbf{m}, \mathbf{c})}\right] Z_t \tag{6.19}$$

Still, the vanishing line should be computed, and a reference length should be used. Knowing three non-coplanar segments of the same height, they can be used to compute the vanishing line and Z_t (see Fig. 6.10). In the experiments carried out, the firewalls and special beacons with known height where used (see Fig. 6.11). However, as height reference, trees or other natural landmarks could be used as well.

Using this procedure, the estimated height in pixels for any point of the contour on the image $h_{j,t}(s)$ can be transformed into an estimation of the height in world coordinates.

Figure 6.11: Artificial landmarks considered for the height estimation of the flames.

Figure 6.12: Aerial images of a fire gathered from an helicopter. These images are usually affected by vibrations.

6.7 Obtaining measurements from aerial robots

In the case of UAVs, the camera pose can be computed by using the sensors on board. As in the case of Chapters 3 and 4, the cameras are calibrated beforehand, using the method of Zhang (2000). Then, the position of each cell in the grid can be associated to a pixel on the image plane, as explained in Section 4.2.1.

As commented in Section 6.4, monitoring the details of fire implies to analyze the dynamic evolution of the fire front and flames on the image plane. The feature extraction method described requires motionless cameras. Considering UAVs with hovering capabilities, unavoidable control errors, turbulence and vibrations produce changes in the camera position which lead to image motion (Fig 6.12 shows an example of aerial images of a fire). Therefore, it is necessary to solve the camera motion problem. Currently, electro-mechanic systems can be used to cancel vibrations, but the systems of this kind are heavy, expensive and usually have a residual vibration. Image processing procedures can be used for software-based image motion estimation and cancelation.

6.7.1 Image stabilization

Image stabilization consists of canceling the background scene motion induced by the motion of the camera. This can be achieved if the apparent motion between consecutive images is computed. The

algorithms for image motion computation fall into two broad classes:

- The algorithms of the first class try to obtain a full *dense motion field* for all the pixels on the image. Then, it is straightforward to use this motion field to warp the images to a common frame, thus eliminating the induced motion. These techniques are usually based on spatio-temporal filtering methods over the sequence of images (Fleet, 1992; Farnebäck and Nordberg, 2002). The main drawback of these kind of methods is that they require a high overlap among consecutive images (equivalently a high frame rate compared to the camera motion).

- The techniques of the second class estimate a *sparse* image motion field. Only the motion of salient features on the image plane is computed. Then, this sparse motion field is used to estimate a model of the full image motion. Finally, the model is applied on all pixels to warp the image and eliminate the background motion between the current image and the previous one. All in all, these methods can cope with larger inter-image motion, with a smaller overlap among image, although, as a global image motion model is considered there are certain situations in which these methods are less advisable.

For images gathered by an UAV, the motion and vibrations of the vehicle will induce large motions on the image plane. Also, the frame rate may be restricted due to bandwidth limitations in the case of low cost UAVs, that may not have enough processing power on-board for image processing and that therefore should relay the images to a computer on ground. Then, a method of the second class will be used for image motion estimation. This method is described next.

6.7.2 Description of the feature matching algorithm

The sparse motion field is usually computed by obtaining matches between persistent features on the image plane. There are different kind of features that can be used, such as corners, lines or blobs. Point-like features present some advantages in unstructured environments over other possibilities (Ferruz and Ollero, 2000; Jung and Lacroix, 2001).

In general, the point features selected in these methods correspond to the centre of regions that present a good 2D structure in the image. The structure of an image I can be measured using several techniques (Johansson and Farnebäck, 2002). One of them is the *gradient or structure tensor* (Bigün and Granlund, 1987; Harris and Stephens, 1988):

$$\mathbf{T}(\mathbf{m}) = (\nabla I(\mathbf{m}))(\nabla I(\mathbf{m}))^T * G \tag{6.20}$$

where $*$ is the convolution operator and G is usually a low-pass Gaussian kernel. Here, the method presented by Ferruz and Ollero (2000) is used. In this case, those points $\mathbf{m}$ on the image plane for which the lowest eigenvalue of $\mathbf{T}$ is over a given threshold present a good structure and are selected as interest points (Harris and Stephens, 1988; Shi and Tomasi, 1994). The features finally selected are small image patches around these points (Fig. 6.13). These features are then tracked along the sequence of images, using as a similarity measure such as the Sum of Squared Differences or Normalized Correlation over the pixels values of the image patch.

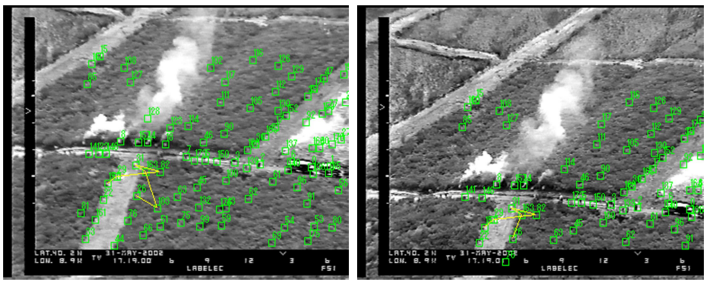

Figure 6.13: Interest points extracted and tracked in two consecutive images.

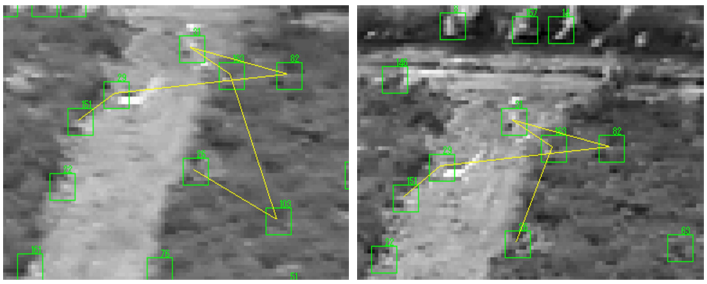

Figure 6.14: Clusters of points are used as persistent features.

The original method has been improved by Ferruz, and now, clusters of points are also used as persistent features to be tracked (Fig. 6.14). In this way, the features present invariant properties under certain transformations, improving the robustness of the matching between them (Merino et al., 2007b).

As a result of the matching procedure, for any given pair of images, a set of matches $\{\mathbf{m}_{t-1}^{[k]}, \mathbf{m}_t^{[k]}\}, k = 1, \ldots, N$ is obtained.

6.7.3 Image motion model

Assuming that an UAV with hovering capabilities is used, the camera vibrations can be assimilated as small pure rotations. Considering a coordinate frame centered on the camera position at time $t-1$, following (3.57) the homogeneous pixel coordinates $\mathbf{m}_{t-1}$ corresponding to a point $\mathbf{p}$ are:

$$s\mathbf{m}_{t-1} = \mathbf{A}\begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{p}\\ 1\end{pmatrix} = \mathbf{A}\mathbf{p} \tag{6.21}$$

If the camera only rotates, at time t the same point is imaged at pixel $\mathbf{m}_t$, with coordinates:

$$s'\mathbf{m}_t = \mathbf{A}\begin{pmatrix}\mathbf{R}_t & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{p}\\ 1\end{pmatrix} = \mathbf{A}\mathbf{R}_t\mathbf{p} \tag{6.22}$$

The calibration matrix $\mathbf{A}$ is invertible, so from (6.21) $\mathbf{p}$ can be expressed as $\mathbf{p} = s\mathbf{A}^{-1}\mathbf{m}_{t-1}$. Therefore,

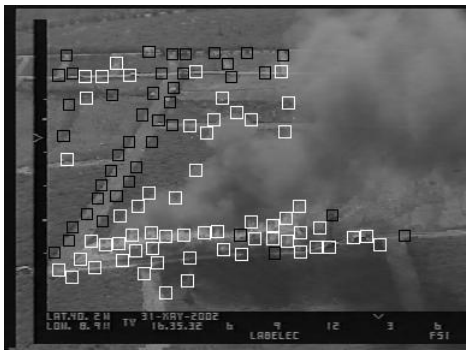

Figure 6.15: Typical aerial close-range image of a fire and features selected. It can be seen how some interest points correspond to moving objects (smoke).

$$s''\mathbf{m}_t = \mathbf{A}\mathbf{R}_t\mathbf{A}^{-1}\mathbf{m}_{t-1} = \mathbf{H}_\infty\mathbf{m}_{t-1} \tag{6.23}$$

The 3×3 matrix $\mathbf{H}_\infty = \mathbf{A}\mathbf{R}_t\mathbf{A}^{-1}$ is called the *infinity homography* (Hartley and Zisserman, 2004). This homography is used as a model for the image motion under vibration. As the homography is defined up to an scale factor, it has only 8 degrees of freedom. The sparse motion field obtained from the feature matching algorithm will be used to estimate the most likely homography between two consecutive images.

6.7.4 Homography computation

From the sparse motion field given by the set of matches obtained, the objective is to infer the homographic motion model for the full image. From (6.23), each match $\{\mathbf{m}_{t-1}^{[k]}, \mathbf{m}_t^{[k]}\}$ adds a restriction on the homography:

$$s^{[k]}\mathbf{m}_t^{[k]} = \mathbf{H}_\infty\mathbf{m}_{t-1}^{[k]} \tag{6.24}$$

Since $\mathbf{H}$ has only eight degrees of freedom, four matches are needed to determine $\mathbf{H}_\infty$ linearly. In practice, more than four correspondences are available, and the overdetermination is used to improve accuracy. Appendix C presents a linear algorithm for the computation of the homography from a set of pair correspondences.

There are additional issues that should be considered when obtaining the model. The model derived in (6.23) is only valid if the points $\mathbf{p}^{[k]}$ corresponding to the matches $\{\mathbf{m}_{t-1}^{[k]}, \mathbf{m}_t^{[k]}\}$ do not move, that is, it is only valid for an static scene. However, in the case of aerial images of a forest fire, smoke and even the fire itself are moving objects that can occupy wide areas of the image (see Fig. 6.15). In order to compute an accurate model, these objects with *independent motion* in the scene should be detected and not considered when computing $\mathbf{H}_\infty$. These objects will be treated as outliers.

The homography computation is summarized in Fig. 6.16. The computation is divided in two basic steps: *outlier rejection* and *robust estimation*. The first step tries to detect the outliers in order

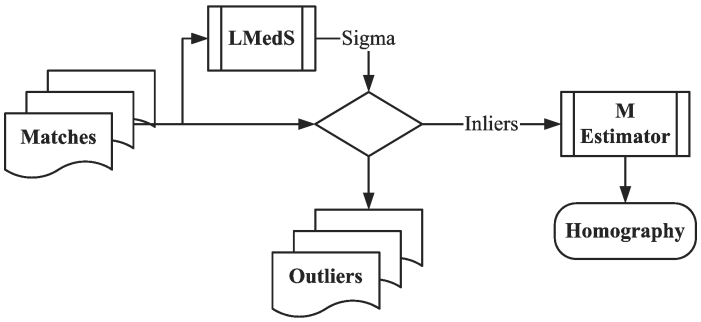

Figure 6.16: Homography computation diagram.

Figure 6.17: Warping problem representation.

to increase the accuracy of the computed homography. The overall design of the outlier rejection procedure used here is based on Least Median of Squares (LMedS) (Zhang, 1995).

In the second step, *robust estimation*, the homography is computed by using an iterative method (M-Estimator) (Zhang, 1997) that allows to automatically weight the set of data as a function of the residue in each iteration. A Fair function was selected because the set of data were well filtered by the outlier rejection stage. The Fair M-Estimator will guarantee a good convergence to the correct solution in few iterations (Ollero et al., 2004).

6.7.5 Image warping

The homography computed is used as a model for the full image motion. This model allows to relate the position of any pixel $\mathbf{m}_t$ of image I_t to its corresponding position $\hat{\mathbf{m}}_{t-1} = \mathbf{H}_{\infty}^{-1}\mathbf{m}_t$ in the previous image I_{t-1}. Combining consecutive transformations is possible to warp all the images to a common frame, compensating the camera motion.

Warping each pixel using this transformation can be used to compensate the image motion induced by the camera motion. In general, the transformed position $\hat{\mathbf{m}}_{t-1}$ will not correspond to an integer position due to the algebraic operations. It is necessary to define a method to assign an integer position to the transformed pixels in order to warp the images correctly. This problem can be seen in Figure 6.17 where the arrow marks the position in the right image where the pixel in the left image has been transformed. The integer nature of the pixels force to select one of the four neighbors (highlighted with a square).

Therefore, a method based on *pixel similarity* is employed. The technique uses the Euclidean

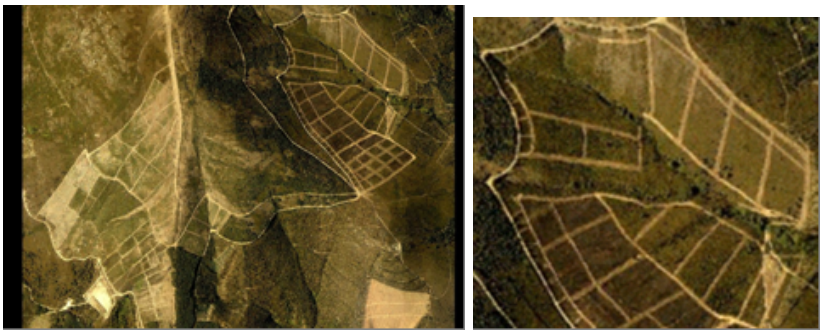

Figure 6.18: An aerial view of the plots employed during the fire experiments at Gestosa, Portugal. The right image shows a close up view of one of the zones, in which the plot side is around 75 m.

Figure 6.19: Two pictures of the actual deployment of the system. Only the frontal cameras are shown. Cameras in other places are also deployed (transmitting their images to the central station).

distance between the RGB value of the actual pixel and the values of the four neighbors in the previous image (pixels signaled in Figure 6.17). The final position of the pixel will be the one corresponding to the neighbor with the shortest colour distance. Thus, for each pixel, the method tries to minimize the RGB differences with respect to the previous image. This helps to increase the alignment between sequenced images, even correcting little errors in the homography computation (Caballero et al., 2004).

6.7.6 Height estimation

Aerial images are not used for height estimation, as the image stabilization procedure induces distortions on the image height, due to the parallax effect. Although this distortion is related to the height of the flames, and therefore could be considered for this estimation, currently the height is computed only by the static cameras on ground.

6.8 Description of the experiments and experimental results

The techniques described have been validated in a high number of controlled forest fire experiments carried out in Serra da Gestosa (Portugal). In these field experiments, square plots of up to 150×100 meters were burned with controlled safety conditions. Figure 6.18 shows an aerial image of the Gestosa experimental site. The rectangular plots burned in the experiments can be observed in the images. The experiments mobilized significant resources including 80 firemen and 5 fire-fighters trucks. A more detailed description can be found in (Viegas et al., 2002).

Two types of stations were deployed in the experiments: several sensing stations and one processing station, where the algorithms run in a robust portable computer. The deployment of the stations is strongly dependent on the topography of the terrain. At least two sensing stations, one frontal and one lateral, are deployed. In some cases more stations can be used, and in some experiments a helicopter provides aerial views. Typically, frontal stations contain one infrared and one visual camera and lateral stations contain one visual camera. The processing station is usually located close to the frontal sensing station to facilitate the acquisition of the images. Video transmitters are used for distant cameras. Figure 6.19 shows the deployment of the frontal cameras for two different experiments.

6.8.1 Calibration of the likelihood functions

The setup of the system consists of deploying the sensors and calibrating relations (6.15) and (6.19) for the static cameras from landmarks seen on the images. These relations are used to estimate the likelihood of the measurements. In this section, an analysis of the validity of these likelihood functions is performed.

Figure 6.20 shows two different views of one of the plots during the experiments. The coordinates of the firewalls that can be distinguished were measured using DGPS. Some points on these firewalls are used as landmarks. For the determination of the relation $\mathbf{H}_j$ for each static camera, the calibration procedure uses correspondences between pixels on the image plane and the landmarks. The algorithm of Section C.1 is used to compute the homography and an estimation of the errors in this homography, in the form of a covariance matrix $\Sigma_{\mathbf{H}}$, for a given error level in the pixel position (set by the user).

These errors in the homography induce errors in the pixel positions corresponding to the cells of the grid, and are considered in the estimation process. Figure 6.20 shows a validation analysis, in which Gaussian errors of 0 mean are added to the pixel coordinates corresponding to the landmarks. Then, the resultant homography is used to estimated the projection on the image plane of the central point of the plot. A comparison of the estimated variances with those obtained by using a Monte Carlo analysis, and with an standard linearization procedure, is shown.

It can be seen that the UT is more stable that the corresponding first-order linearization. However, for high errors, the uncertainty can be underestimated, and then in the filters the uncertainties on the pixel position are conservatively overestimated. Figure 6.21 shows the estimated mean position of the pixel. It can be seen that no bias is introduced.

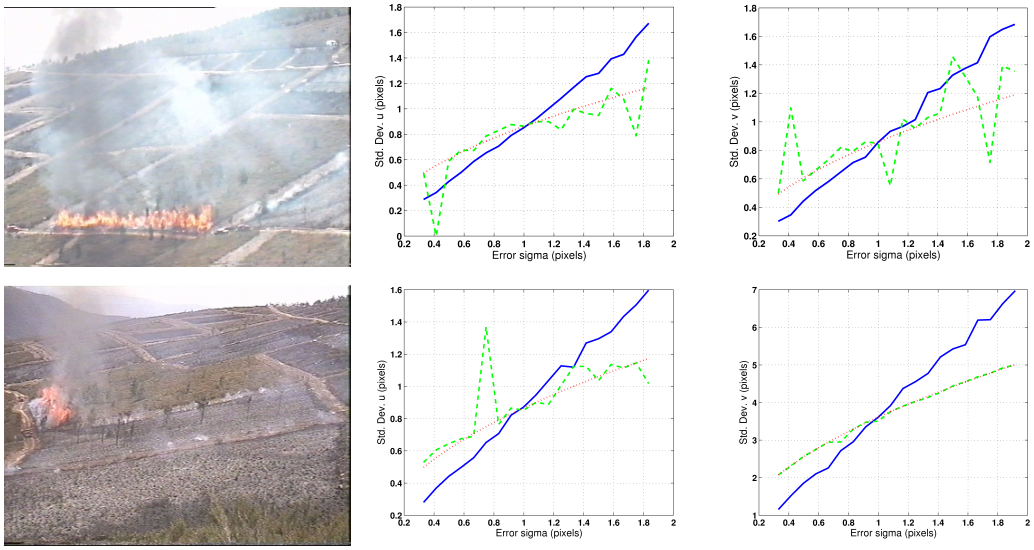

Figure 6.20: The errors on the estimated homography **H** will induce errors on the estimated pixel position for the cells of the grid. This is estimated using the UT. The figure shows a comparison of the estimated standard deviations with a Monte Carlo analysis and a first-order estimation based on linearization. The estimated position of the central point of the plot is considered. The plot is about 120×80 meters, and the image is 384×288. Solid: Monte Carlo. Dashed green: linearization. Dashed: Unscented Transform. Top: frontal view. Bottom: lateral view.

As commented in Section 6.6.1, the relation is invertible and can be used to determine the position of the pixels on the reference plane. The same analysis has been carried out in this case. Figure 6.22 shows the results of this analysis for two images (frontal and lateral) in the same fire experiment. An overlay on the images represents the geo-referenced X and Y coordinates. The dimensions of the plot are 120x80 m. In this and the previous figures it can be seen how the different views complement each other, as depending on the orientation with respect to the fire, the errors are higher in some coordinates.

The height calibration procedure for planar scenes employed with the static cameras has also been analyzed. On the field, several landmarks of known height (3 meters) were deployed. An example of these landmarks was shown in Fig. 6.11. Three of them are used to compute the vanishing line, and the reference height Z_c. Then, using these parameters the height of the fourth landmark (also of 3 meters) is obtained from its image. A Monte Carlo analysis is made adding errors to the height in pixels for the reference landmarks. Figure 6.23 shows the results. The resultant distribution can be well approximated by a Gaussian.

In (Criminisi et al., 2000), closed-form expressions for an estimation of the variances based on a first-order Taylor expansion of (6.18) can be found. Here, again the UT is employed to obtain an estimation of the errors on the height from the errors of the estimated vanishing line and point. Figure 6.23 shows a comparison of the variance estimated by the UT and that estimated in a Monte Carlo analysis.

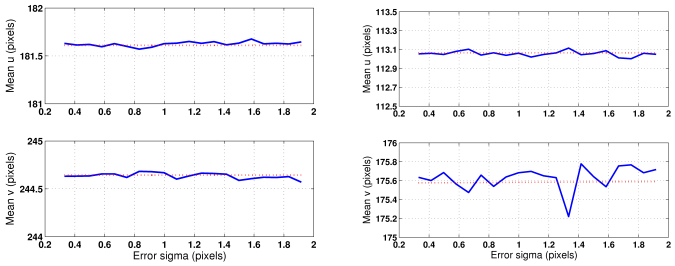

Figure 6.21: The figure shows the estimated mean position. Solid: Monte Carlo. Dashed: Unscented Transform. Left: frontal view. Right: lateral view.

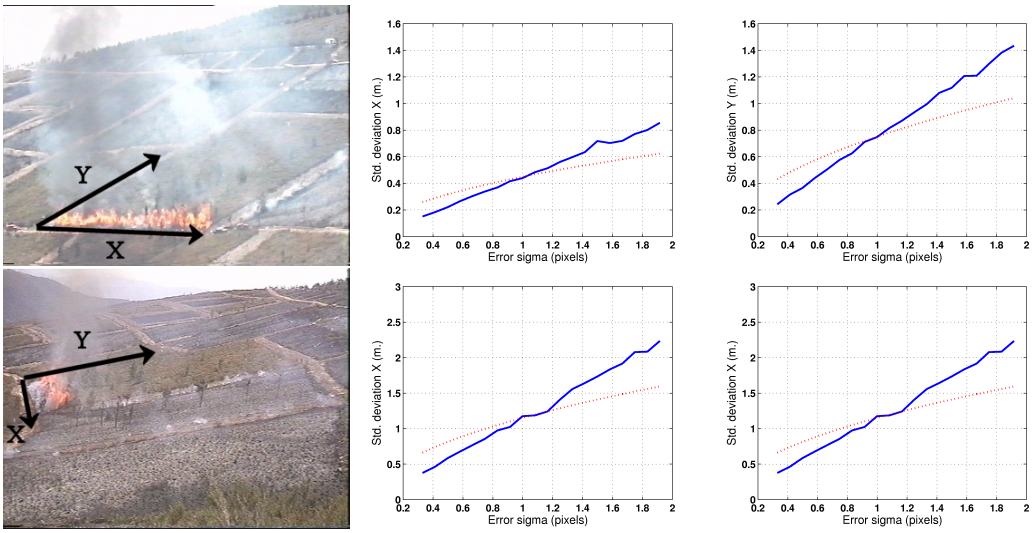

Figure 6.22: Monte Carlo analysis of the geo-location procedure. Solid: Monte Carlo samples. Dashed: Unscented Transform.

6.8.2 Results

This section presents some results obtained using the system for several fire experiments. Most of these experiments consist of setting fire on one of the plots of Fig. 6.18. The fires are lit with different configurations, depending on the interest of the researchers on fire behavior characterization.

In some of the experiments, threads were deployed on the plots at fixed intervals in order to validate some of the measurements obtained (see Fig. 6.24). These threads are used to know at what time instants the fire reaches the respective distances, and thus to obtain an estimation of the evolution of the most advanced point of the fire front.

Plot 345

The first experiment shows results for a plot in which threads were deployed following the scheme described in Fig. 6.24. In this experiment, a linear front evolves uphill from the lower part of a

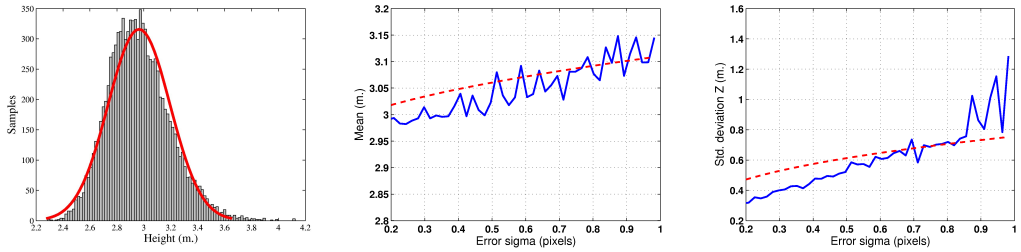

Figure 6.23: Monte Carlo analysis of the height computation procedure. Left, histogram obtained compared to a Gaussian distribution for a given error level on the computation of the vanishing line. Center and right: mean and standard deviation obtained by using the UT (dashed) compared to the Monte Carlo analysis (solid) for different error levels.

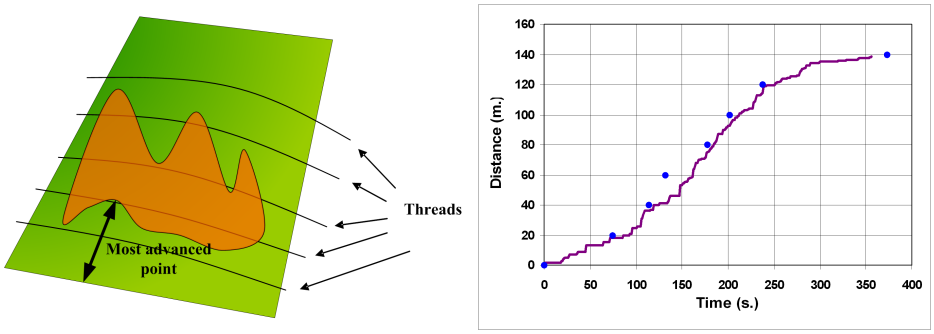

Figure 6.24: Left: Threads were deployed in some plots for validation purposes. One thread is placed all several meters, and the time each thread is cut by the fire is annotated. Right: Distance of the most advanced point of the fire front with respect to the lower firewall for Plot 345. The dots correspond to time instants and distances at which the threads deployed were cut by the fire.

plot. Figure 6.24 also shows the estimated distance of the fire with respect to the lower firewall. This distance is calculated obtaining the fire front as described above and determining its maximum distance with respect to the firewall (whose coordinates are known).

Moreover, Fig. 6.25 shows the evolution of the fire front shape each 30 seconds, and compares it to the estimated front shape obtained manually by a posterior analysis of photographs gathered from a helicopter.

Plot 522

Figure 6.26 shows several IR images of the Plot labeled as 522 during the experiments. In this case, the fire is initiated in two separated fronts that evolve until they meet at the end of the experiment.

Figure 6.27 shows the estimated evolution of the fire front shape, compared to the estimated position of the fire obtained from the analysis of photographs employing photogrammetric techniques. Qualitatively, it can be seen how the evolution of the two separated fronts is captured.

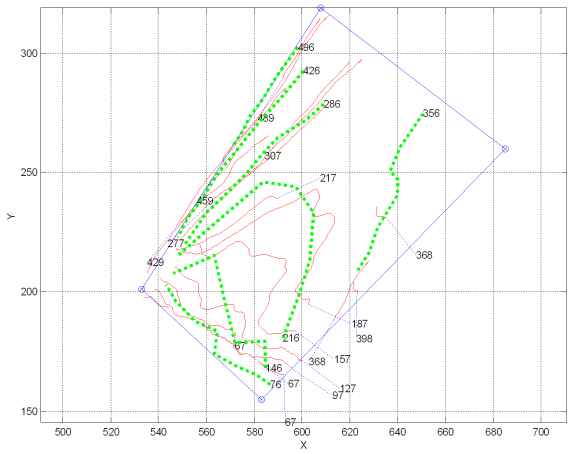

Figure 6.25: Fire front shape evolution (each 30 seconds) as computed by the algorithm (solid) and compared to the fire front obtained by photogrammetric techniques (dashed).

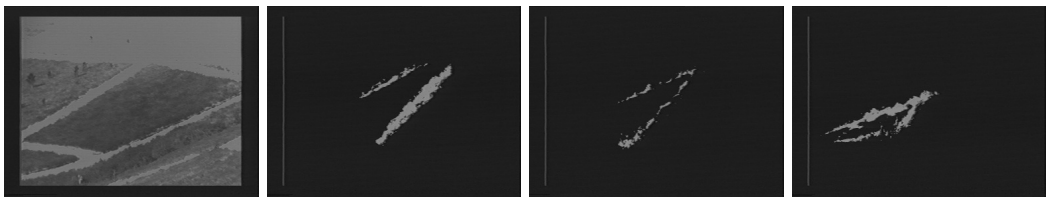

Figure 6.26: Plot 522. Fire is set in two linear fronts that advance one over the other.

Plot 062

Again, in this experiments two fronts are set up, on evolving uphill and the other downhill. The image of Fig. 6.1 corresponds to this experiment. Fig. 6.28 shows the evolution of the estimated fire front.

Also in this case, the estimated maximum height of the flames is shown in Fig. 6.29. The estimated height qualitatively follows the intensity of the fire, reaching its maximum when both fronts meet. The estimated height is sometimes affected by smoke.

Plot 520

In this experiment, a plot of approximately 10000 square meters is burnt. A conventional helicopter is used to gather aerial images. Also, a frontal and a lateral views from static cameras are available. A linear fire front is ignited on the lower part of the plot, and the fire propagates uphill.

Figure 6.30 shows several frames gathered from the helicopter, after the image stabilization and feature extraction algorithms are applied. It can be seen how the stabilization algorithm effectively removes the motion induced by the helicopter.

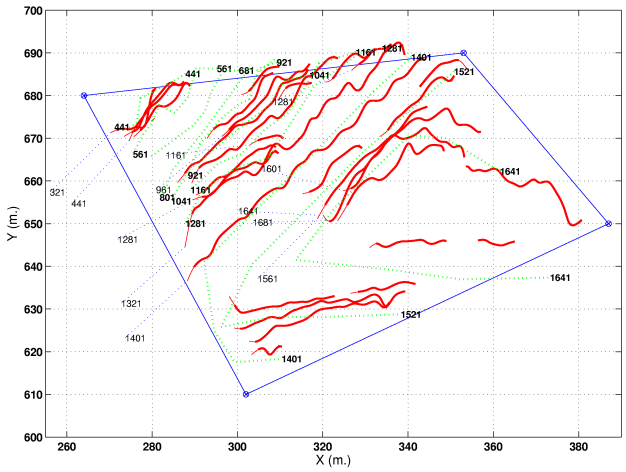

Figure 6.27: Fire front shape evolution for plot 522 as computed by the algorithm (solid) and compared to the fire front obtained by photogrammetric techniques (dashed).

From these features, the fire probability grid is evolved. Figure 6.31 shows the estimation of the fire front shape each 20 seconds, compared to an estimation based also on images. Figure 6.32 shows the evolution of the most advanced point of the fire front.

Also, during the general experiments of the COMETS project, actual small-size UAVs were tested on fire monitoring activities, but of small-size fires. These experiments demonstrated the capabilities of cooperative fire monitoring. The final results in this experiments are shown in Chapter 8, where they are included in a more general fire fighting mission.

6.9 Conclusions

This chapter has presented cooperative perception algorithms applied to the problem of monitoring forest fires. A system that can integrate images from static cameras and cameras on board aerial vehicles has been presented. The system has been tested in actual experiments involving controlled fires. In these experiments, images from static cameras and cameras on board helicopters are considered.

One of the main conclusions of the chapter is that it is feasible to develop automatic systems for forest fire perception. Moreover, the experiments involving aerial images show that UAVs can be very helpful for fire fighting activities like fire monitoring, as they can cover the gap between the scales given by systems based on satellites and those based on cameras on towers.

Besides, the system allows incorporating data from several UAVs, which definitely is very convenient as views from different vantage points can be obtained. Moreover, in the case of large fires the full fire front can be covered. Also, the presence of smoke can be prevented by moving the vehicles adequately. The use of several UAVs for cooperative fire monitoring is further illustrated in the

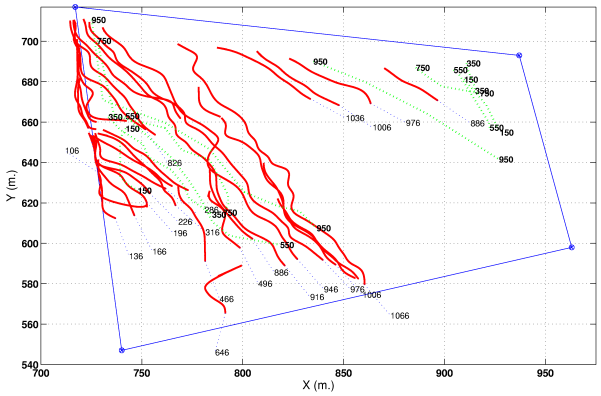

Figure 6.28: Fire front shape evolution for plot 062 as computed by the algorithm (solid) and compared to the fire front obtained by photogrammetric techniques (dashed).

experiments of Chapter 8.

Although the tested system is centralized, the techniques described in Chapter 4 can be used to decentralize the grid estimations, which would make the system scalable to a higher number of cameras and UAVs.

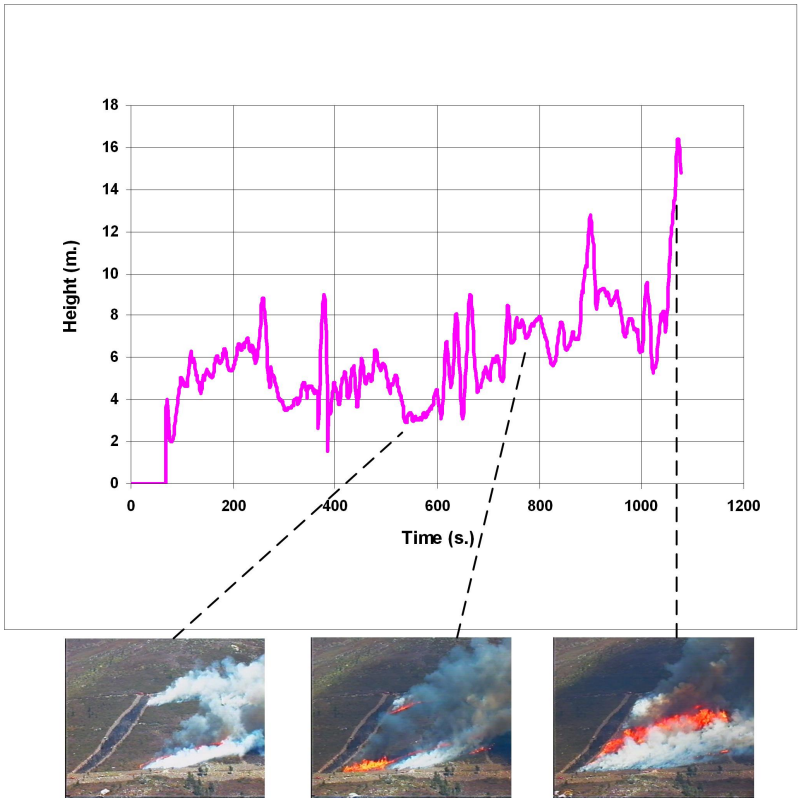

Figure 6.29: Evolution of the maximum height for the experiment 062.

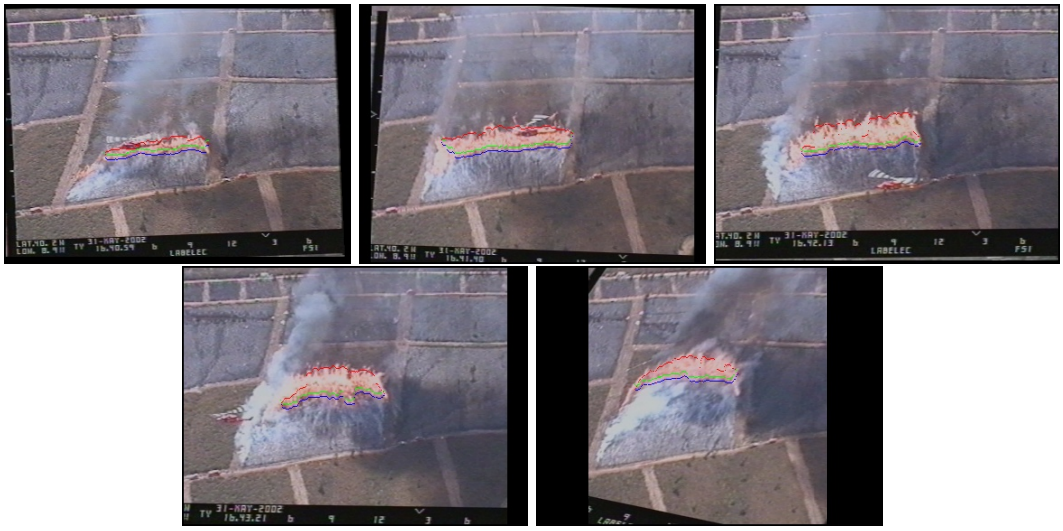

Figure 6.30: Frames 3, 44, 77, 144 and 219 of the aerial sequence gathered during the fire experiment 520. The extracted fire front contour is also shown.

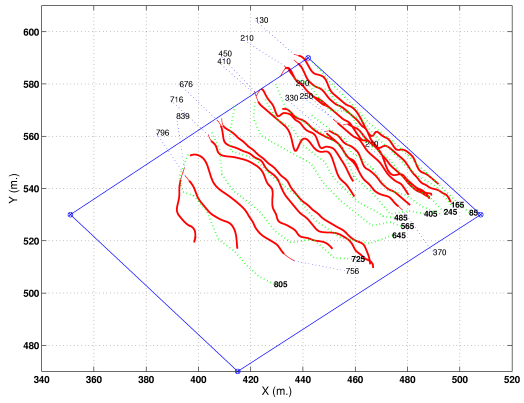

Figure 6.31: Evolution of the fire front for plot 520 estimated from the images gathered by the helicopter (solid) and compared with the front obtained with an static camera (dashed).

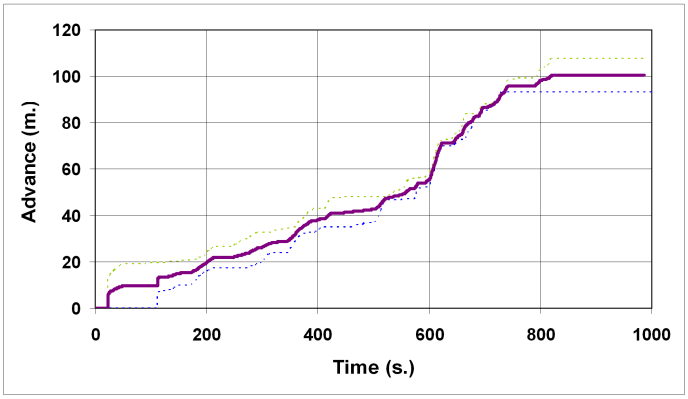

Figure 6.32: Evolution of the most advanced point of the fire front with respect to the lower firewall for experiment 520. Estimations from the aerial and frontal static cameras (dashed) and combined estimation (solid).

And then of course I've got this terrible pain in all the diodes down my left side.

Marvin, in Douglas Adams' *The Hitchhiker's Guide to the Galaxy (1979)*

Chapter 7

A visual odometer for UAV motion estimation Application to multi-UAV fleets

In the previous chapters, the UAVs are assumed to be localized in a global frame. This is accomplished by integrating the measurements provided by GPS receivers and IMUs that the vehicles carry on board. However, GPS can be subject to errors and inaccuracies given by satellite occlusion or multi-path signal reception. Moreover, these data cannot be used for relative precise positioning with respect to objects, such as building walls or a landing platform.

This chapter presents a computer vision-based method to estimate the motion of one UAV. This method computes the real motion from the homographies that relate several views of a planar scene. As it will be shown, several views of the same plane allow extracting the real camera motion up to a scale factor, which can be disambiguated by using a range sensor.

Furthermore, as pointed out in the previous chapters, in multi-robot applications it is needed to know the relative position of the robots in order to share information. The previously presented cooperative schemes assume that the UAVs can estimate their positions in a global common reference frame. In the general case of having local positioning systems, a mechanism is required for establishing relations between the local coordinates of each robot. It will be shown that, under certain circumstances, vision can be used to compute the relative motion among UAVs.

7.1 Related work on vision-based UAV motion estimation and navigation

For small-size UAVs, the most common exteroceptive sensor used for UAV navigation is vision. Range sensors are not very common in these small UAVs, and are mainly used for landing (such as laser or ultrasonic range finders).

Vision is used as a positioning sensor in several UAV projects. It is argued that some applications that need a very accurate relative position of the UAV with respect to objects, as close proximity

flight, cannot rely on inertial navigation systems or global positioning systems (Amidi, 1996). Thus, vision is used as a method to sense relative position.

The concept of visual odometer (Amidi et al., 1999) was implemented in the autonomous helicopter developed at Carnegie-Mellon University (CMU). By using this concept, the helicopter can visually lock-on to ground objects and sense relative helicopter position and velocity in real-time. Template matching and stereo techniques are used to estimate the displacement of the object between consecutive images. This, combined with angular rate sensing (provided by gyros), allows to estimate the UAV motion. In this approach, only a few templates are tracked. Each cycle new targets are selected to avoid losing targets. The same visual tracking techniques, combined with inertial sensors, have been applied to autonomously take off, following a prescribed trajectory, and landing. The CMU autonomous helicopter also demonstrated autonomous tracking capabilities of moving objects by using only on-board specialized hardware.

The use of vision for autonomous landing has been actively researched. In the early nineties, Dickmanns and Schell (1992) presented some results of the possible use of vision for landing an airplane. Zhang and Hintz (1995) developed a video based attitude and height sensor for low altitude airborne vehicles. Neural networks are used to compute attitude and height from the measures obtained by a video camera that observes the structured light pattern created by a set of infrared lasers. Yakimenko and others (2002) consider the problem of determining the position and orientation of an aircraft with respect to a ship from the images of several points taken by an infrared camera. All these methods relay on artificial beacons that can be recognized on the images.

In the BEAR project at Berkeley, vision-based pose estimation of unmanned helicopters relative to a landing target and vision-based landing of an aerial vehicle on a moving deck are researched (Shakernia et al., 2002; Vidal et al., 2002a). Results on multiple view geometry are used to impose constraints over multiple views of a planar landing target with a single camera. These constraints, summarized in the so called multiple view matrix, allow to compute the displacement of the UAV given three or more views of the target. The idea is similar to that presented here, although the method is different. The feature extraction and matching are simplified by using an artificial target of known shape and size. Also, in that approach, the known size of the landing target allows to obtain the scale of the displacement. Using off-the-shelf hardware, the vision system in the BEAR project is able to operate at 30 Hz. Vision is situated in the control loop as a high level supervisor which sends the current position of the landing target to the navigation controller.

Computer vision is also used for safe landing in the AVATAR project. Thus, Garcia-Pardo and others (2001) present a strategy and an algorithm relying on image processing techniques to search the ground for a safe place. Vision-based techniques for landing on a helipad of known shape are presented in (Saripalli et al., 2003; Saripalli and Sukhatme, 2003), where also the case of landing on a slow moving helipad is considered. The system applies target recognition using computer vision (through threshold-based segmentation and by using the moments of inertia of a helipad of known shape) and a Kalman filter for target tracking. In (Saripalli et al., 2002) they also present a technique for helicopter position estimation using a single CMOS camera pointing downwards and with a large field of view, and a laser pointer to project a signature onto the surface below in such a way that

it can be easily distinguished from other features on the ground. The search for a safe place for landing by using vision has been also considered by other researchers, as in (Johnson et al., 2005) at JPL, or in (Meingast et al., 2004; Bosch et al., 2006).

CSIRO in Australia has performed research in vision systems for UAV navigation, and Corke et. al (2000) have analyzed the use of stereo vision for height estimation in small size helicopters.

In Georgia Tech, vision-based aided navigation for UAVs has been considered. Thus, in (Wu et al., 2005) the authors present an Extended Kalman Filter that combines GPS measurements with image features obtained from a known artificial target for helicopter position estimation. Also, in (Proctor and Johnson, 2005) a vision-only method for landing an UAV is presented. Artificial beacons signaling the landing zone are perceived by the onboard camera and used to determine the relative position of the vehicle. Some results in simulation are presented. For a recent summary of the ongoing work in this lab, including vision-aided inertial position estimation, target search and tracking see (Ludington et al., 2006).

Simultaneous Localization and Mapping (SLAM) techniques are closely related to the position estimation problems. SLAM is now a well known technique and has been used mainly with ground robots. Among them, only a few are mainly based on vision sensors. Also, there are some applications of vision for SLAM with UAVs.

SLAM techniques for aerial robotics have been developed at LAAS. The perception system being designed for the Karma airship (Lacroix et al., 2002; Hygounenc et al., 2004) applies stereo vision, interest point matching and Kalman filtering techniques for simultaneous localization and mapping using stereo vision. The robust interest point matching algorithm combined with stereo vision is used for blimp motion estimation. Some of the interest points are used as landmarks in the SLAM approach for the filter update phase. By using stereo vision, a dense range map can be obtained. Karma carries a stereo bench 2-meters wide (placed along the blimp gondola), and is able to build maps with an accuracy of 10 cm. and a cell resolution of 5 square centimeters. However, this approach is not suitable for helicopters, as the baseline of the stereo rig that can be carried is small, and therefore it limits the height at which the UAV can fly.

UAV simultaneous localisation and map building with vision using a delta fixed wind platform is also presented in Kim and Sukkarieh (2004). Artificial landmarks of known size are used in order to simplify the landmark identification problem. The known size of the landmarks allows to use the camera as a passive range/bearing/elevation sensor. In this work, images of known landmarks and inertial data are used as ground truth and they show the corrective properties of the SLAM algorithm. Initial work on the use of vision-based bearing-only SLAM in UAVs is presented in (Lemaire et al., 2005).

Related to the work of vision-based navigation is the direct use of image measurements in the control loop of an UAV, that is, the use of *visual servoing* techniques for UAV control. In this case, instead of using vision to estimate UAV position and then feeding this position into a controller visual features are directly used as reference. And thus, in (Zhang and Ostrowski, 1999), a feedback linearization-based controller for small indoor blimps, considering the system dynamics and using images from a colored ball as inputs, is presented. Similar ideas are presented in (Fukao et al.,

2003). Both approaches deal with small blimps in indoor environments, and the processing is done in computers on ground.

In (Hrabar and Sukhatme, 2003), the use of an omnidirectional camera for controlling an helicopter is considered. The camera is used to maintain the helicopter in the centroid of a set of artificial targets. The processed images are directly used to command the helicopter. The paper shows the feasibility of the procedure, but no actual control is tested.

In (Hrabar et al., 2005), optical flow and stereo vision are used for reactive navigation of an UAV in urban-like environments. Optical flow applied over sideways-looking cameras is used to maintain the UAV at a safe distance from side obstacles. Depth measurements from a forward-facing stereo pair are used to avoid front obstacles. The paper shows an interesting configuration in which only two fisheye cameras are used, and some actual experiments with the AVATAR helicopter are presented.

Del Cerro et al. (2002) present the use of stereo vision to estimate the relative position of power cables for their inspection with an UAV. In the work of Mejías et al. (2006), vision is used to track features of buildings. The image features and GPS measurements are combined together into a controller to maintain the UAV aligned with the selected features. Control design and stability analysis of image-based controllers for aerial robots are presented in (Mahony and Hamel, 2005). Proctor et al. (2006) present recent work on vision-based control of an fixed wing aircraft.

Regarding multi-UAV localization, much less work can be identified. In (Ling et al., 2003), an architecture for multi-vehicle SLAM is studied for its use with UAVs. The paper deals with the issues of data association and communication, and some simulation results are presented.

7.2 A visual odometer based on planar homographies

The objective is then to obtain the motion of one UAV (displacement and changes in attitude) from images obtained from cameras on board. Monocular cameras are considered. It is also important to obtain an estimation of the errors committed in the estimation, which is something usually not considered in most of the approaches presented above.

The method infers the displacement of the camera from the motion of point features tracked along the sequence of images. This corresponds in part to the Structure from Motion (SfM) problem in computer vision. Many approaches have been proposed for the solution of the problem in the general 3D case, such as the 5 point algorithm (Nistér, 2004; Nistér et al., 2004). Here, an assumption is made that allows simplifying the problem: the scene is assumed to be planar. This situation is typical for UAVs flying in outdoor scenarios. It is usually a valid approximation when the altitude with respect to the ground is big compared to the variations in the ground height. Moreover, planar structures appear in building inspection and urban scenarios, or during landing. Also, planar structures usually pose ill-condition challenges to SfM algorithms that try to recover the full 3D structure of the scene.

Any SfM method recovers the motion up to an scale factor, and so does the method presented here. However, as it will be seen, the distance to the reference plane is sufficient to disambiguate the scale. In the case of UAVs, range-finder sensors can be used for this.

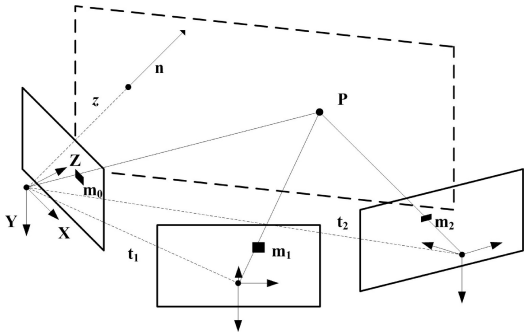

Figure 7.1: Multi-view geometry for a planar scene.

7.2.1 The projective geometry of two views of a planar scene

A 3-D plane with unit normal vector $\mathbf{n}_0$ is the set of points $\mathbf{p}_0$ that accomplish:

$$\mathbf{n}_0^T \mathbf{p}_0 = d \tag{7.1}$$

where d is the distance of the plane to the origin of the reference frame. If a second reference frame is considered (corresponding to a second camera), any point $\mathbf{p}_0$ in the first reference frame is expressed as:

$$\mathbf{p}_1 = \mathbf{R}_{01}(\mathbf{p}_0 - \mathbf{t}_{01}) \tag{7.2}$$

in the second reference frame, where $\mathbf{t}_{01}$ is the relative displacement of the second point of view expressed in the first camera coordinate frame, and $\mathbf{R}_{01}$ is the rotation matrix that transforms a vector in the first camera coordinate frame into a vector expressed in the second camera coordinate frame.

For the points $\mathbf{p}_0$ belonging to the plane, the expression can be expressed in different terms. As $\mathbf{n}_0^T \mathbf{p}_0 / d = 1$, then:

$$\mathbf{p}_1 = \mathbf{R}_{01}(\mathbf{p}_0 - \mathbf{t}_{01}\mathbf{n}_0^T \mathbf{p}_0/d) = \mathbf{R}_{01}(\mathbf{I} - \mathbf{t}_{01}\mathbf{n}_0^T/d)\mathbf{p}_0 \tag{7.3}$$

This special form of the transformation has also implications on the way the images of these points are transformed. The image of the points of the plane is given by the pin-hole model described by (3.57). Using homogeneous pixel coordinates:

$$s_0\mathbf{m}_0 = \mathbf{A}\begin{pmatrix}\mathbf{I} & \mathbf{0}\end{pmatrix}\begin{pmatrix}\mathbf{p}_0 \\ 1\end{pmatrix} = \mathbf{A}\mathbf{p}_0 \tag{7.4}$$

From (7.3), the image of the same points in the second view is:

$$s_1\mathbf{m}_1 = \mathbf{A}\mathbf{R}_{01}\begin{pmatrix}\mathbf{I} & -\mathbf{t}_{01}\end{pmatrix}\begin{pmatrix}\mathbf{p}_0 \\ 1\end{pmatrix} = \mathbf{A}\mathbf{R}_{01}(\mathbf{I} - \mathbf{t}_{01}\mathbf{n}_0^T/d)\mathbf{p}_0 \tag{7.5}$$

Therefore,

$$s_1\mathbf{m}_1 = s_0\mathbf{A}\mathbf{R}_{01}(\mathbf{I} - \mathbf{t}_{01}\mathbf{n}_0^T/d)\mathbf{A}^{-1}\mathbf{m}_0 \tag{7.6}$$

that is, there is a linear relation (in homogeneous coordinates) that relates two views, 0 and 1, of the same planar scene (see Fig. 7.1). This linear relation (a homography, see Appendix C, and Section 6.7.3) can be expressed as:

$$\mathbf{H}_{01} = \mathbf{A}\cdot\mathbf{R}_{01}\cdot(\mathbf{I} - w\mathbf{t}_{01}\cdot\mathbf{n}_0^T)\cdot\mathbf{A}^{-1} \tag{7.7}$$

with $w = 1/d$. These homographies are called plane-induced homographies. The infinity homography of Section 6.7.3 is a special case, when there is only a rotation ($\mathbf{t}_{01} = \mathbf{0}$) or when the plane is located at infinity ($d \to \infty$).

7.2.2 Estimation of motion from planar homographies

For UAV navigation, the objective is to derive the motion of the camera given the relation $\mathbf{H}_{0i}$ that relates the current image and a reference frame. Next sections will show how $\mathbf{H}_{0i}$ can be computed from image correspondences.

If the camera is calibrated ($\mathbf{A}$ is known), then it is possible to obtain two valid solutions for the motion of the camera $\mathbf{R}_{0i}$, $\mathbf{t}_{0i}$, and the plane normal $\mathbf{n}_0$ from $\mathbf{H}_{0i}$, up to a scale factor; for instance using the *SVD algorithm* described in (Triggs, 1998) or by using the *four point planar algorithm for two views* described in (Faugeras, 1993). Only the product $w\mathbf{t}_{0i}\cdot\mathbf{n}_0^T$ can be recovered, and then, the vector $\mathbf{t}_{0i}$ is normalized so that $\|\mathbf{t}_{0i}\| = 1$, and the distance to the plane $w_{0i} = 1/d$ is obtained in (the current) baseline units. The SVD method is summarized next.

SVD algorithm

Let $\mathbf{H}_c = \mathbf{A}^{-1}\mathbf{H}_{0i}\mathbf{A}$ be the *calibrated* or Euclidean version of the homography. Therefore,

$$\mathbf{H}_c = \mathbf{R}_{0i}\cdot\underbrace{(\mathbf{I} - w_{0i}\mathbf{t}_{0i}\cdot\mathbf{n}_0^T)}_{\mathbf{H}} = \mathbf{R}_{0i}\mathbf{H} \tag{7.8}$$

The obtention of the motion parameters is based on the SVD of $\mathbf{H}_c = \mathbf{U}_c\mathbf{D}\mathbf{V}^T$ Therefore, the inputs to the algorithm are the matrices $\mathbf{U}_c = \begin{pmatrix}\mathbf{u}_{c1} & \mathbf{u}_{c2} & \mathbf{u}_{c3}\end{pmatrix}$, $\mathbf{V} = \begin{pmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3\end{pmatrix}$ and $\mathbf{D} = \mathrm{diag}(d_1, d_2, d_3)$ (with $d_1 \geq d_2 \geq d_3$).

The SVD of $\mathbf{H}_c$ and $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^T$ are identical but for $\mathbf{U}_c = \mathbf{R}_{0i}\mathbf{U}$, with $\mathbf{U} = \begin{pmatrix}\mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3\end{pmatrix}$

The following relations allow the obtain the displacement parameters from the SVD:

$$\mathbf{H}\cdot(\mathbf{t}_{0i}\wedge\mathbf{n}_0) = \mathbf{t}_{01}\wedge\mathbf{n}_0 \tag{7.9}$$

$$\mathbf{H}^T\cdot(\mathbf{t}_{0i}\wedge\mathbf{n}_0) = \mathbf{t}_{0i}\wedge\mathbf{n}_0 \tag{7.10}$$

$$\mathbf{H}\cdot\mathbf{t}_{0i} = \mathbf{t}_{0i} - (w\mathbf{n}_0^T\mathbf{t}_{0i})\mathbf{t}_{0i} = (1 - w\mathbf{n}_0^T\mathbf{t}_{0i})\mathbf{t}_{0i} \tag{7.11}$$

where $\wedge$ denotes the cross product. That is, if the singular values d_j of $\mathbf{H}$ are distinct, $\mathbf{t}_{0i} \wedge \mathbf{n}_0$ is a left and right singular vector of $\mathbf{H}$. Moreover, $\mathbf{t}_{0i}$ is an eigenvector of $\mathbf{H}$ with eigenvalue $1 - w_{0i}\mathbf{n}_0^T\mathbf{t}_{0i}$.

Assume that $\mathbf{t}_{0i} \wedge \mathbf{n}_0$ corresponds to the singular vectors $\mathbf{v}_2$ and $\mathbf{u}_2$. As the homography is defined up to an scale factor, the singular values are normalized so that $\mathbf{D} = \mathrm{diag}(d_1/d_2, 1, d_3/d_2)$.

As the columns of $\mathbf{V}$ are orthonormal, $\mathbf{v}_1$ and $\mathbf{v}_3$ should span the subspace that contains $\mathbf{n}_0$ and $\mathbf{t}_{0i}$. Therefore, for some α and β,

$$\begin{aligned} \mathbf{n}_0 &= \beta\mathbf{v}_1 - \alpha\mathbf{v}_3 \\ \mathbf{n}_0 \wedge (\mathbf{t}_{0i} \wedge \mathbf{n}_0) &= \alpha\mathbf{v}_1 + \beta\mathbf{v}_3 \end{aligned} \tag{7.12}$$

and $\alpha^2 + \beta^2 = 1$ (as $\mathbf{n}_0$ is an unitary vector). As $\|\mathbf{n}_0 \wedge (\mathbf{t}_{0i} \wedge \mathbf{n}_0)\| = \|\mathbf{H}_c[\mathbf{n}_0 \wedge (\mathbf{t}_{0i} \wedge \mathbf{n}_0)]\|$, then $\alpha^2 + \beta^2 = (d_1\alpha)^2 + (d_3\beta)^2$, and therefore $(\alpha, \beta) \propto (\pm\sqrt{1 - d_3^2}, \pm\sqrt{d_1^2 - 1})$. Should $\mathbf{t}_{0i} \wedge \mathbf{n}_0$ had corresponded to $\mathbf{v}_1$ or $\mathbf{v}_3$, there would have been no real solution for (α, β).

A similar argument considering $\mathbf{t}_{0i}$ and the left vectors $\mathbf{u}_1$ and $\mathbf{u}_3$ leads to obtain that $\mathbf{R}_{0i}\mathbf{t}_{0i} = -\beta\mathbf{u}_{c1} - \alpha\mathbf{u}_{c3}$.

On the other hand, $\mathbf{t}_{0i}$ is an eigenvector of $\mathbf{H}$ with eigenvalue $1 - w\mathbf{n}_0^T\mathbf{t}_{0i}$. Therefore $\mathbf{H}_c\mathbf{t}_{0i} = (1 - w_{0i}\mathbf{n}_0^T\mathbf{t}_{0i})\mathbf{R}_{0i}\mathbf{t}_{0i}$, and thus $\mathbf{t}_{0i}$ is proportional to $\mathbf{H}_c^{-1}(\mathbf{R}_{0i}\mathbf{t}_{0i})$.

$$\mathbf{t}_{0i} \propto \mathbf{V}\mathbf{D}^{-1}\mathbf{U}_c^T(-\beta\mathbf{u}_{c1} - \alpha\mathbf{u}_{c3}) \Rightarrow \mathbf{t}_{0i} \propto \beta/d_1\mathbf{v}_1 + \alpha/d_3\mathbf{v}_3 \tag{7.13}$$

The actual vector $\mathbf{t}_{0i}$ can be recovered, as $\|\mathbf{t}_{0i}\| = 1$ by definition. After that, the value of w_{0i} can be obtained from the eigenvalue relation (7.11) as $w_{0i} = d_1 - d_3$.

Finally, recovering $\mathbf{R}_{0i} = \mathbf{U}_c\mathbf{U}^T$ requires to recover $\mathbf{U}$. As $\mathbf{v}_2 = \mathbf{u}_2$, then $\mathbf{u}_1$ and $\mathbf{u}_3$ should be in the space spanned by $\mathbf{v}_1$ and $\mathbf{v}_3$.

$$\begin{aligned} \mathbf{u}_1 &= \lambda\mathbf{v}_1 + \zeta\mathbf{v}_3 \\ \mathbf{u}_3 &= \zeta\mathbf{v}_1 - \lambda\mathbf{v}_3 \end{aligned} \tag{7.14}$$

Once $\mathbf{t}_{0i}$ and w are known, λ and ζ can be recovered again from (7.11), so $(\lambda, \zeta) \propto (1 + d_1d_3, \pm\alpha\beta)$. The final equations are given by Algorithm 7.1 (Triggs, 1998).

Obtaining a unique solution

The algorithm, then, provides two valid solutions $\mathbf{R}_{0i}^{[1,2]}$, $\mathbf{t}_{0i}^{[1,2]}$ and $\mathbf{n}_{0i}^{[1,2]}$ for the motion parameters (with the translation up to a scale factor) from the homography $\mathbf{H}_{0i}$. Given more than two images and the homographies that relate each image i with the reference frame, it is possible to obtain an unique solution because the plane normal $\mathbf{n}_{0i}$ must be the same for all the relations $\mathbf{H}_{0i}$ (it is the plane normal expressed in the reference frame).

Then, the correct solution should be computed for all the images. In (Shakernia et al., 2002) a method to compute the correct solution based on the uniqueness of the normal $\mathbf{n}_0$ is proposed. Given the set of possible normals:

Algorithm 7.1 $(\mathbf{t}_{0i}^{[1,2]}, \mathbf{R}_{0i}^{[1,2]}, \mathbf{n}_{0i}^{[1,2]}, w_{0i}) \leftarrow$ Motion_from_homog$(\mathbf{H}_{0i})$

1: $[\mathbf{U}, \mathbf{D}, \mathbf{V}] = \text{svd}(\mathbf{H}_{0i})$
2: $\mathbf{D} = \text{diag}(d_1, d_2, d_3)$
3: $d_1 \leftarrow d_1/d_2, d_3 \leftarrow d_3/d_2$
4: $a = \sqrt{1 - d_3^2},\ b = \sqrt{d_1^2 - 1}$
5: $a \leftarrow \frac{a}{\sqrt{a^2+b^2}},\ b \leftarrow \frac{b}{\sqrt{a^2+b^2}}$
6: $c = 1 + d_1 d_3\ d = \sqrt{1 - d_3^2}\sqrt{d_1^2 - 1}$
7: $c \leftarrow \frac{c}{\sqrt{c^2+d^2}},\ d \leftarrow \frac{d}{\sqrt{c^2+d^2}}$
8: $e = -b/d_1,\ f = -a/d_3$
9: $e \leftarrow \frac{e}{\sqrt{e^2+f^2}},\ f \leftarrow \frac{f}{\sqrt{e^2+f^2}}$
10: $\mathbf{t}_0^{1[1]} = e\mathbf{v}_1 + f\mathbf{v}_3$
11: $\mathbf{t}_0^{1[2]} = e\mathbf{v}_1 - f\mathbf{v}_3$
12: $\mathbf{R}_{01}^{[1]} = \mathbf{U} \begin{pmatrix} c & 0 & d \\ 0 & 1 & 0 \\ -d & 0 & c \end{pmatrix} \mathbf{V}^T$
13: $\mathbf{R}_{01}^{[2]} = \mathbf{U} \begin{pmatrix} c & 0 & -d \\ 0 & 1 & 0 \\ d & 0 & c \end{pmatrix} \mathbf{V}^T$
14: $\mathbf{n}_0^{[1]} = b\mathbf{v}_1 - a\mathbf{v}_3$
15: $\mathbf{n}_0^{[2]} = b\mathbf{v}_1 + a\mathbf{v}_3$
16: $w_{0i} = d_1 - d_3$

$$S_n = \{\mathbf{n}_{01}^{[1]}, \mathbf{n}_{01}^{[2]}, \mathbf{n}_{02}^{[1]}, \mathbf{n}_{02}^{[2]}, \mathbf{n}_{03}^{[1]}, \mathbf{n}_{03}^{[2]}, \ldots\} \tag{7.15}$$

where the superindex indicates the two possible normal solutions and the subindex is the image number sequence. This method defines an empirical threshold ϵ, and considers as the correct solution the normal $\mathbf{n}_{01}^{[1]}$ or $\mathbf{n}_{01}^{[2]}$ that allows to find a sequence of normals that accomplish:

$$\|\mathbf{n}_{01}^{[1,2]} - \mathbf{n}_{0i}^{[1,2]}\| \leq \epsilon \quad \forall i > 1 \tag{7.16}$$

where ϵ must be carefully tuned to guarantee a proper algorithm operation. If the value of ϵ is too high then both solutions can accomplish the constraint and it is not possible to detect the correct one. On the other hand, if ϵ is too low then no solution will accomplish the constraint.

A more reliable method is proposed to detect the correct solution. Thus, if $\mathbf{n}_{01}^{[1]}$ and $\mathbf{n}_{01}^{[2]}$ were correct, there would be two set of solutions, S_{n^1} and S_{n^2}. The uniqueness of the normal leads to the following constraints:

$$\|\mathbf{n}_{01}^{[1]} - \mathbf{n}_{0i}^{[j(i)]}\| \leq \epsilon_1 \quad \forall \mathbf{n}_{0i}^{[j(i)]} \in S_{n^1} \tag{7.17}$$

$$\|\mathbf{n}_{01}^{[2]} - \mathbf{n}_{0i}^{[j(i)]}\| \leq \epsilon_2 \quad \forall \mathbf{n}_{0i}^{[j(i)]} \in S_{n^2} \tag{7.18}$$

where ϵ_1 and ϵ_2 are the minimal values that guarantee an unique solution for equations (7.17) and

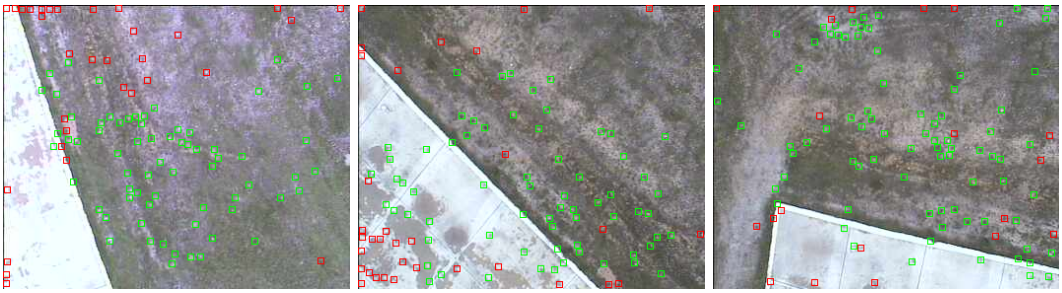

Figure 7.2: Matches in three images during an helicopter flight.

(7.18) respectively. The pairs $\{S_{n^1}, \epsilon_1\}$ and $\{S_{n^2}, \epsilon_2\}$ are computed separately by means of the following iterative algorithm (only the case of ϵ_1 is described):

1. The distance between $\mathbf{n}_{01}^1$ and the rest of normals of S_{n^1} is computed.
2. ϵ_1 is set to an initial value.
3. For the current value ϵ_1, check if there exists an unique solution.
4. If no solution is found, increase the value of ϵ_1 and try again with step 3). If multiple solutions were found, decrease ϵ_1 and try again with step 3). If an unique solution was found, then finish.

The correct solution is chosen between both options as the one that achieves the minimum ϵ.

Another option is to employ an initial estimation of the helicopter attitude and assume an approximate plane orientation, which can be used to disambiguate the correct estimation (for instance, for a downward looking camera, the plane normal can be assumed parallel to the camera axis).

Once the correct $\mathbf{n}_0$ is known, it is possible to obtain $\mathbf{R}_{0i}$ and $\mathbf{t}_{0i}$ for any image.

Solving the scale factor

One important aspect is that the translation $\mathbf{t}_{0i}$ is computed up to an scale factor ($w_{0i}\|\mathbf{t}_{0i}\|$), as mentioned before. If an independent sensor is used, such as a range sensor, an altimeter, GPS, etc., the scale factor can be obtained by measuring the distance from the first view to the plane ($1/w$).

In the next sections, it is assumed that this distance is measured independently.

7.2.3 Estimation of the planar homography

Section 6.7.4 described a procedure to obtain the homography that relates two consecutive images $\mathbf{H}_{(i-1)i}$ from the matches between point features (see Fig. 7.2 for a typical aerial view and the resultant matches). The same procedure can be used here. From this homography, it is possible then to estimate the camera motion between consecutive images.

Moreover, given the linear nature of the homographic transformations in homogeneous coordinates, it is possible to express the image motion from image n to the reference frame 0, $\mathbf{H}_{0n}$, as a

composition of the relative motion between the intermediate images. If T_I is a sequence of n images of a planar scene:

$$T_I = \{I_0, I_1, I_2..., I_n\} \tag{7.19}$$

and the $n-1$ homography matrices that relate these images are known:

$$\mathbf{H}_{01}, \mathbf{H}_{12}, \mathbf{H}_{23}..., \mathbf{H}_{(n-1)n} \tag{7.20}$$

then, the transformation from image n to k will be defined by:

$$\mathbf{H}_{kn} = \prod_{i=k}^{n-1} \mathbf{H}_{i(i+1)} \tag{7.21}$$

Thus, it is possible to compose the complete motion between any image of the sequence and the reference frame by means of the relative motion computation. Applying the algorithm of the previous section, the motion of the camera with respect to the first frame can then be obtained.

7.2.4 Error estimation

Errors in the homography

It is important to obtain an estimation on the errors committed on the computation of the UAV motion. These errors arise from several sources. The main source is due to errors in the position of the matches, which translate into errors of the estimated homography. Section C.2 describes how to obtain an estimation of the covariances of the errors when estimating the homography between two consecutive images, $\mathbf{H}_{(i-1)i}$. Therefore, each time a new homography is obtained, the covariance $\mathbf{\Sigma}_{(i-1)i}$ of $\mathbf{h}_{(i-1)i}$ (the homography in vector form) is also estimated.

Following (7.21), this homography can be composed with the previous ones to obtain the global homography that relates image i and the reference frame 0. In matrix form, $\mathbf{H}_{0i} = \mathbf{H}_{(i-1)i}\mathbf{H}_{0(i-1)}$. If the errors of the current homography $\mathbf{H}_{0(i-1)}$ (represented by $\mathbf{\Sigma}_{0(i-1)}$) and the new local relation $\mathbf{H}_{(i-1)i}$ are independent, then the covariance of the resultant homography $\mathbf{\Sigma}_{0i}$ can be obtained using a first order approximation as:

$$\mathbf{\Sigma}_{0i} = \frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{0(i-1)}} \mathbf{\Sigma}_{0(i-1)} \frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{0(i-1)}}^T + \frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{(i-1)i}} \mathbf{\Sigma}_{(i-1)i} \frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{(i-1)i}}^T \tag{7.22}$$

It is necessary to obtain the Jacobians of this equation expressed in vector form. If $\{h_{ij}\}$ are the components of the local homography $\mathbf{H}_{(i-1)i}$, then (see Section C.3):

$$\frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{(i-1)i}} = \begin{pmatrix} \mathbf{H}_{0(i-1)}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{0(i-1)}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}_{0(i-1)}^T \end{pmatrix} \tag{7.23}$$

Figure 7.3: The Hero helicopter in building inspection operation.

$$\frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{0(i-1)}} = \begin{pmatrix} h_{11}\mathbf{I} & h_{12}\mathbf{I} & h_{13}\mathbf{I} \\ h_{21}\mathbf{I} & h_{22}\mathbf{I} & h_{23}\mathbf{I} \\ h_{31}\mathbf{I} & h_{32}\mathbf{I} & h_{33}\mathbf{I} \end{pmatrix} \tag{7.24}$$

Errors in the motion parameters

The errors committed in the homography $\mathbf{H}_{0i}$ and in the distance to the reference plane $(1/w)$ will introduce errors in the estimated displacement $\mathbf{R}_{0i}$ and $\mathbf{t}_{0i}$ when applying Algorithm 7.1. The Unscented Transform is used, and then, from the covariances $\mathbf{\Sigma}_{0i}$ of the errors on the estimated homography $\mathbf{H}_{0i}$ and the estimated variance of the distance to the plane, an estimation on the mean and covariances of the computed pose can be derived by applying Algorithm 3.5.

Another potential option would be to obtain a first order estimation of the covariances of the errors, by linearizing the operations in the algorithm. The algorithm involves solving the SVD of $\mathbf{H}_{0i}$ and then several operations with its coefficients. However, although the Jacobian of the SVD transformation of a matrix with respect to its elements can be obtained analytically (Papadopoulo and Lourakis, 2000), as commented, the UT provides more accurate results in general, and involves more simple operations.

7.3 Applications to UAV navigation

This section shows some results of the previous algorithm in three situations: UAV motion estimation when moving from one place to another; motion estimation for landing, and relative position estimation for building inspection.

The results shown in this section have been obtained in experiments with three UAVs: The autonomous helicopter Marvin, developed by the Technical University of Berlin, and which will be described in more detail in Chapter 8, the helicopter Heliv (also described in that chapter) and the autonomous helicopter Hero, developed by the University of Seville (Fig. 7.3).

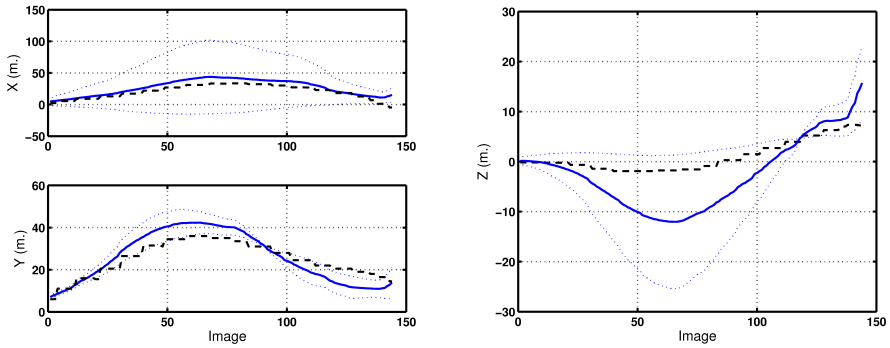

Figure 7.4: Estimated translation (solid) and DGPS estimations (dashed). Left: X and Y coordinates. Right: height. The 3σ bounds are also shown.

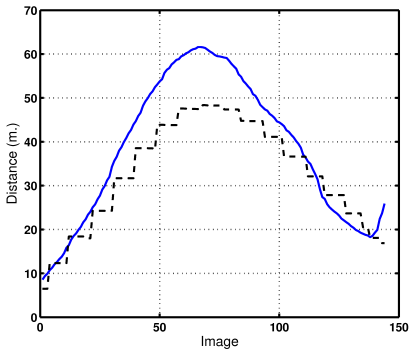

Figure 7.5: Estimated total displacement (solid) and displacement computed from DGPS measures (dashed).

7.3.1 Motion estimation in UAVs

In the first experiment, Heliv is flying following a predefined trajectory. Figure 7.2 shows several images of this flight along with the features tracked. At the same time, the UAV is recording its own attitude and position by means of a differential GPS and a IMU. The camera is mounted on a pan and tilt unit whose angles are recorded as well.

Figure 7.4 shows the estimated translation of the UAV compared to DGPS-based estimates. The reference frame corresponds to the UTM coordinates (X and Y Easting and Northing respectively, Z is the height. The origin corresponds to the first position). The figure also shows the estimated standard deviations on the errors, which are a bit pessimistic, but consistent with the errors committed. Figure 7.5 shows the estimated distance with respect the first frame. The frame rate is about 7 Hz, and therefore the trajectory shown lasts about 20 seconds. It can be seen how the estimated position and variances are consistent with the displacement obtained by the GPS.

Figure 7.6 shows the estimated relative attitude compared to the estimated attitude composing

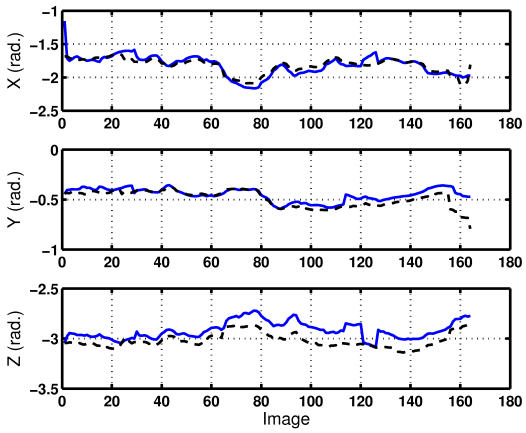

Figure 7.6: Estimated attitude of the camera (solid) and measured from IMU and pan and tilt angles (dashed). The estimated attitude is quite accurate.

the angles provided by the IMU and the pan and tilt unit. The estimated attitude is quite accurate, and more stable than the estimated position.

In a second experiment, the Marvin autonomous helicopter is placed in hover at a given position. The images gathered are used to estimate the motion of the helicopter. Figure 7.7 shows the estimated translation and errors compared to the estimations from a DGPS. It can be seen how the small oscillations during hovering are captured by the estimated motion. Figure 7.8 presents a comparison of the total distance traveled.

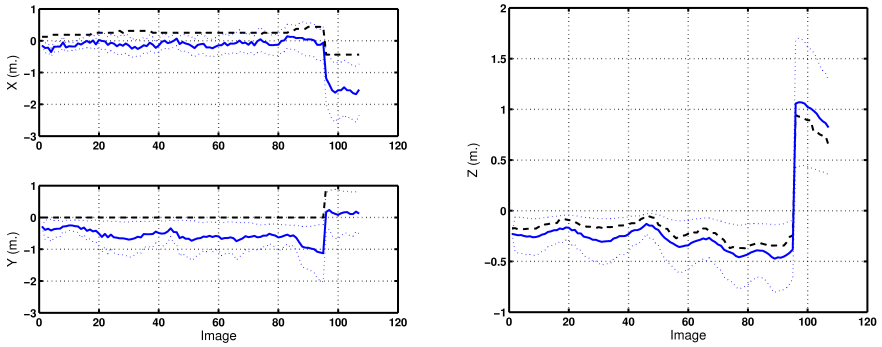

Figure 7.7: Estimated translation (solid) and DGPS estimations (dashed) for a hovering of 20 seconds. Left: X and Y coordinates. Right: height. The 3σ bounds are also shown. It can be seen how the dynamics of the helicopter are captured.

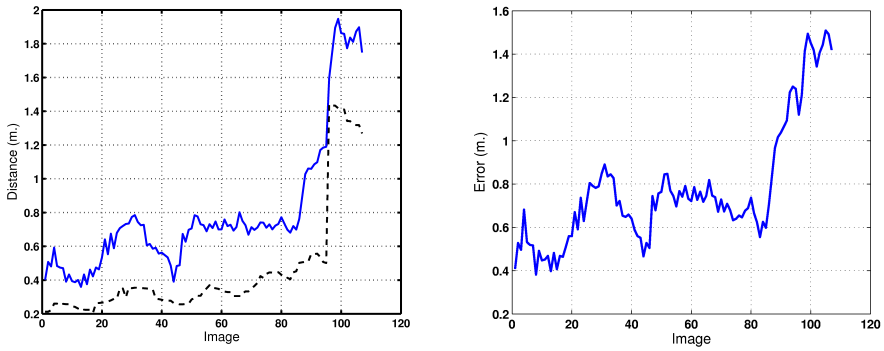

Figure 7.8: Left: estimated distance (solid) and distance computed from DGPS (dashed). Right: Error in the displacement.

Figure 7.9: Matches during a landing sequence.

7.3.2 Motion estimation for landing

The algorithm has been also tested for its potential application to visually-guided landing. The idea is to use vision to estimate the relative position of the UAV with respect to the landing place. Data gathered during an autonomous landing[1] by Marvin has been used.

Figure 7.9 shows three frames of the landing sequence with the obtained matches. It should be pointed out that there are no artificial landmarks for the matching process. Also, in this experiment, the concrete landing platform lacks of structure, which can pose difficulties for the matching procedure. Along the descent, the pan and tilt unit was moving the camera. In this particular experiment, these angles could not be recorded.

Figure 7.10 shows the estimated translation compared with DGPS, along with the estimated errors. The results are very accurate. The technique tends to overestimate the uncertainty, mostly in the altitude. Also, Fig. 7.11 shows the estimated attitude compared to the attitude of the helicopter. The initial configuration of the camera in the landing phase was known and allows to compare the angles to the estimated using an IMU. Along the sequence the tilt angle is varying, and that is the reason for the differences in the X angle that can be seen in the figure.

[1]The autonomous landing was done based on DGPS and ultrasonic sensors.

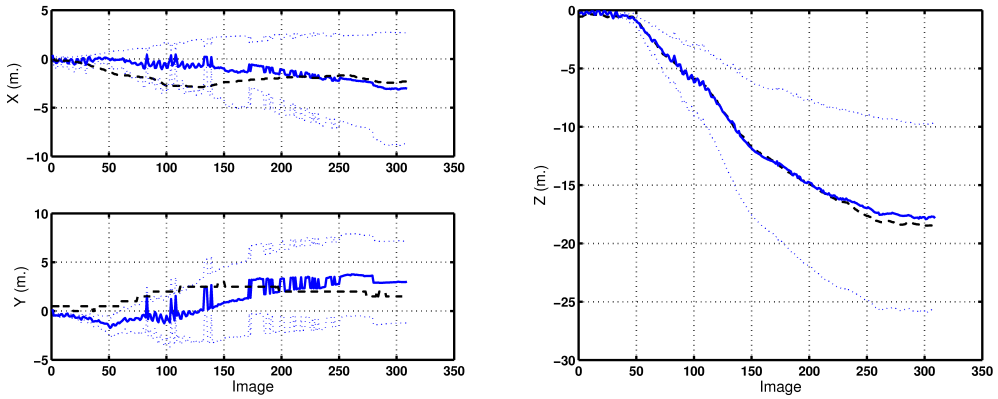

Figure 7.10: Translation during the landing sequence. Visual odometer (solid) and DGPS (dashed). The coordinate system is UTM. The average frame rate is 7 Hz.

7.3.3 Relative motion for building inspections

Another interesting application for UAVs is building inspection. In this application, an UAV should navigate close to a wall, for instance for the detection of thermal leakages. In these situations, the walls can reduce the number of satellites available for the GPS receivers. Also, the GPS signal can worsen due to multi-path reception. On the other hand, it is very interesting to maintain an estimation of the relative pose with respect to the wall, which cannot be obtained from the GPS alone.

Two experiments conducted with Hero are shown here. Figure 7.12 shows three frames of one of the experiments. Figure 7.13 shows the estimated translation obtained by the visual odometer, using the wall as reference plane, again compared to DGPS. The position obtained with the GPS in this case is much less accurate than in the previous experiments, and in fact a visual inspection of the images shows that the visual odometer captures the actual motion of the helicopter more reliably. Also, Fig. 7.14 shows the estimated distance to the wall by the visual odometer.

A second sequence is presented in Fig. 7.15. The estimated translation of the helicopter is shown in Fig. 7.16, whereas the estimated distance to the wall is shown in Fig. 7.17.

In this building inspection application, the same point features have been used for estimating the motion of the UAV. However, in these scenarios, the lack of structure can affect the procedure. A potential solution would be to use different sets of features (for instance, segments) to obtain the image motion.

7.4 Multi-UAV relative position estimation

In the initial chapters, it was stated that the cooperation of several UAVs provides important advantages in certain applications, such as surveillance or precise monitoring of events. The position

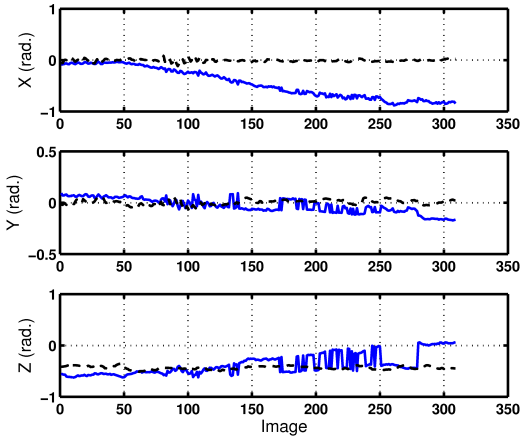

Figure 7.11: Attitude during the landing sequence. Visual odometer (solid) and IMU (dashed).

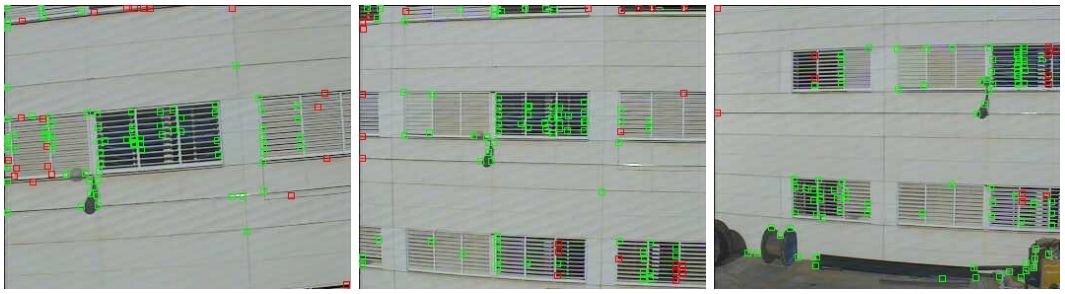

Figure 7.12: Frames 26, 96 and 140 of a building inspection sequence 1.

information of the events should be anchored to the same reference frame in order to share information. Therefore, the computation of the relative transformation between the UAVs local frames is required. In general, GPS receivers are used for positioning, and magnetic compasses and IMUs for attitude. These sensors provide a common reference frame for all UAVs in the fleet. However, in certain applications, other means could be needed to compute this relative pose between companions.

In some multi-robot systems, this problem is solved if the robots are capable of identifying and localizing each other (Konolige et al., 2003). However, it is difficult to extrapolate these techniques to fleets of UAVs since they can not detect each other with simple sensors (UAVs can fly fast and at different altitudes) and in many cases they cannot carry on board the required complex sensors.

Therefore, another approach is considered. If different UAVs identify, using their cameras, common objects in the scene, the relative pose displacement between the UAVs can be computed from these correspondences under certain assumptions. The method described in the previous sections obtains the relative displacement between two views of the same planar scene. The relations presented there also hold if the views are taken by different calibrated cameras. Thus, the relative displacement between two UAVs can be obtained by computing the plane-induced homography matrix between

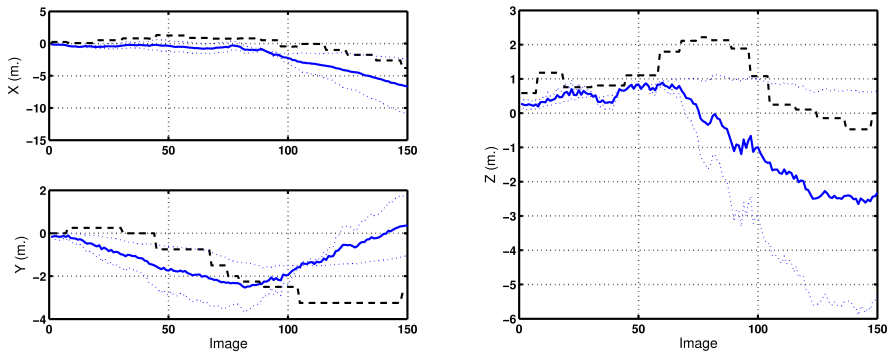

Figure 7.13: Estimated translation for the building inspection sequence 1 (solid) and the estimated translation obtained from the GPS (dashed).

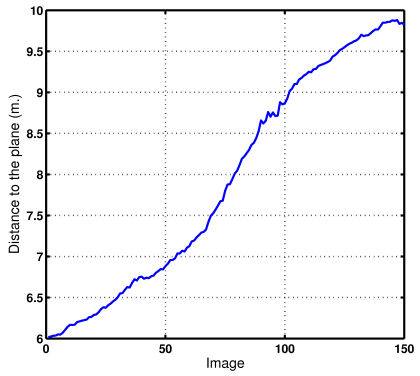

Figure 7.14: Estimated distance to the wall for the building inspection sequence 1.

their cameras.

The main problem is to obtain matches between images gathered by different UAVs to obtain this homography. The features to be matched should be identified in a robust and repeatable way in images from different cameras at different locations and orientations. Point-like features are not suitable for this task, as they do not present enough invariance characteristics. The idea is then to employ other kind of features, more suitable for this application, and to analyze its feasibility.

This last section presents initial results obtained in collaboration with researchers of the University of Linköping, Sweden. A different kind of features, colour segments on images, also called blobs, will be considered. The blob feature extraction and matching techniques are due to Per Erik Forssén[2], and a more detailed description can be found in (Forssén and Granlund, 2003; Forssén and Moe, 2005) and (Merino et al., 2006c). This feature extraction procedure is summarized next.

[2]An implementation and details can be found in http://www.isy.liu.se/~perfo/software/

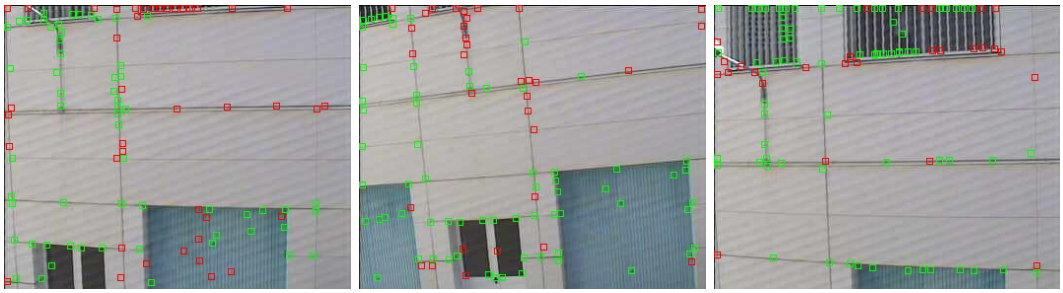

Figure 7.15: Frames 1, 40 and 95 of the building inspection sequence 2.

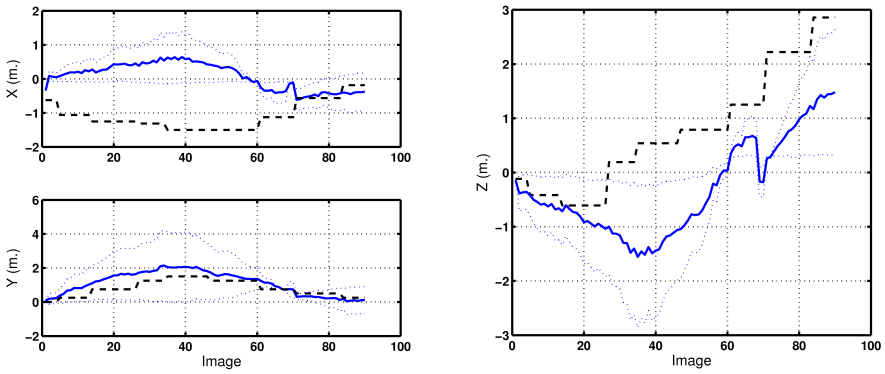

Figure 7.16: Estimated translation for the building inspection sequence 2 (solid) and the estimated translation obtained from the GPS (dashed).

7.4.1 Blob features

Homogeneity features are called blobs in scale-space theory (Lindeberg, 1994). In contrast to segmentation, blob detection does not attempt to segment out exact shapes of objects, but to extract robust and repeatable features, discarding exact shapes and thin connections between patches, see Fig. 7.18. Blob features are also related to maximally stable extremal regions (MSER), (Obdrzálek and Matas, 2002) and to affine invariant neighborhoods, (Tuytelaars and Gool, 2004).

In contrast to (Obdrzálek and Matas, 2002; Tuytelaars and Gool, 2004) the blob features considered here are extracted from colour images, using a clustering pyramid built using robust estimation in local image regions (Forssén, 2004). Each extracted blob is represented by its average color $\mathbf{c}_k$, area a_k, centroid $\mathbf{m}_k$, and inertia matrix $\mathbf{I}_k$; i.e. each blob is a 4-tuple

$$\mathcal{B}_k = \langle \mathbf{c}_k, a_k, \mathbf{m}_k, \mathbf{I}_k \rangle \ . \tag{7.25}$$

Figure 7.19 shows the blob features extracted from an aerial image. These blob features can be seen as a low dimensional representation of the homogeneous colour regions on the image. Figure 7.20 shows the blobs extracted in several images of the same sequence; it can be seen how these blobs are quite stable.

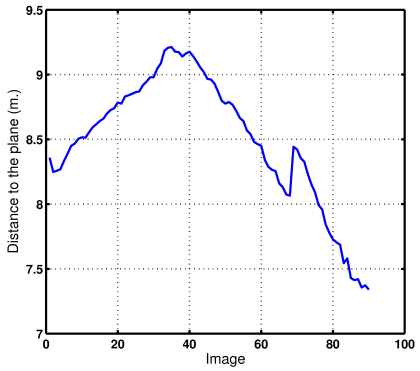

Figure 7.17: Estimated distance to the wall.

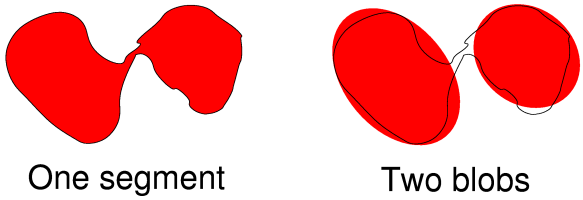

Figure 7.18: Difference between segmentation and blob detection.

7.4.2 Blob-based image matching and homography estimation between images from different UAVs

The procedure of Section 7.2.2 requires to estimate the homography that relates two images gathered from different UAVs. The matching of blobs extracted from images from different points of view is not an easy problem. The approach for solving this problem is to establish relations between blobs as well as between chains of blobs extracted from each image.

The first step is to find initial guesses of correspondences between a blob in one of the images and the blobs in the other, resulting in starting blobs for a set of chains. This is done using only the colour of the blobs. Once these correspondence guesses are made, the rest of the blobs are matched by using a criteria based on the features of the blobs $\mathcal{B}_k$ and also on normalized distances and relative angles between blobs, see Fig. 7.21. This procedure assures that the matching is invariant to scale and rotation.

The result of this step is a set of blob-chain pairs, which can be used as initial hypotheses in the homography estimation process. Typically, in the order of 100 blob features are found in each view. About one quarter of these are successfully matched. The number of blob-chain pairs is in the order of 10. These numbers indicate that the processing time and memory consumption are low.

To facilitate the image matching, a color correction method compensates the colour spectrum of the images to be matched. Colour correction techniques transform the pixel values in an image I to a new image I', such that they appear as if acquired with a different camera, or under different

Figure 7.19: Original image and blob representation.

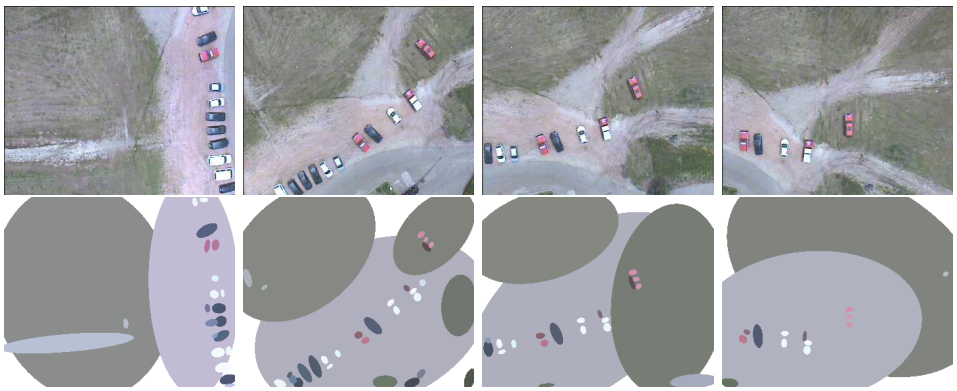

Figure 7.20: Blobs features are quite stable and can be used to obtain matches between images with big changes in the point of view. Top row: original images. Bottom row: blobs extracted.

lighting conditions. This correction can be computed beforehand for the different UAVs. More information on the blob matching can be found in (Forssén and Moe, 2005; Merino et al., 2006c).

The homography estimation procedure is similar to that presented in Section 6.7.4, but in this case, the centroid of the blobs in the blob chains matched is used, instead of the point position of the point features. Again, an outlier rejection method is employed.

The result of a wide baseline matching algorithm on blob features is shown in Fig. 7.22. The left image originates from Heliv with a colour video sensor. The right image originates from Marvin with a high resolution digital camera. The matching blobs are overlaid in both views in Fig. 7.22. As pointed out earlier, finding the homography assumes that the blob features are located on a planar surface. Figure 7.22 shows that this is not always true. However, in practice the procedure can cope with situations when the planar surface requirement is somewhat relaxed. This means that the distance from the camera to the ground should be fairly large compared to the height differences on the ground. If this is the case, the problem of occlusion in the views is also reduced.

One major problem is to get a sufficiently large common area in the two views, in order to get enough blob features to be able to calculate the correspondences. Since blobs which are crossing the

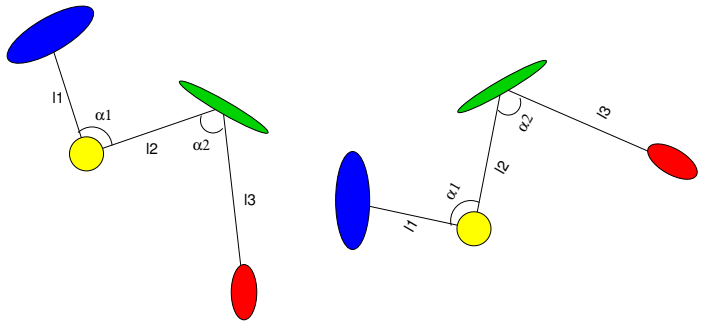

Figure 7.21: Blob chains contain the ratios between consecutive distances and relative angles in the chain. These inter-blob features make the feature vector of the chain rotation and scale invariant.

Figure 7.22: Blobs fulfilling the estimated homography between two images from different UAVs with video sensor and a digital photo camera, respectively.

border of the image cannot be used, there is a requirement that there must be enough distinctive area patches inside the common area of each view. This can be accomplished in different ways; one possibility is to use partial mosaic images to get larger area coverage for the blob matching (Fig. 7.23 shows an example of this approach).

7.4.3 Multi-UAV relative position estimation

Some results have been obtained for images gathered by two UAVs, Marvin and Heliv. In the first experiment presented, the motion of the UAV Marvin is obtained by employing the visual odometer described in the previous sections. This motion is expressed in a local frame of Marvin. Blob matches obtained between the images gathered by both UAVs (Fig. 7.22) are used to estimate the relative motion between Marvin and Heliv. This relative transformation is combined with the Heliv pose (estimated from GPS and IMUs) to obtain the motion of Marvin in the global frame. Also, the relative transformation can be used to estimate the distance to the ground. Figure 7.24 shows the results.

A second experiment was conducted, in this case dividing a full image sequence gathered by the

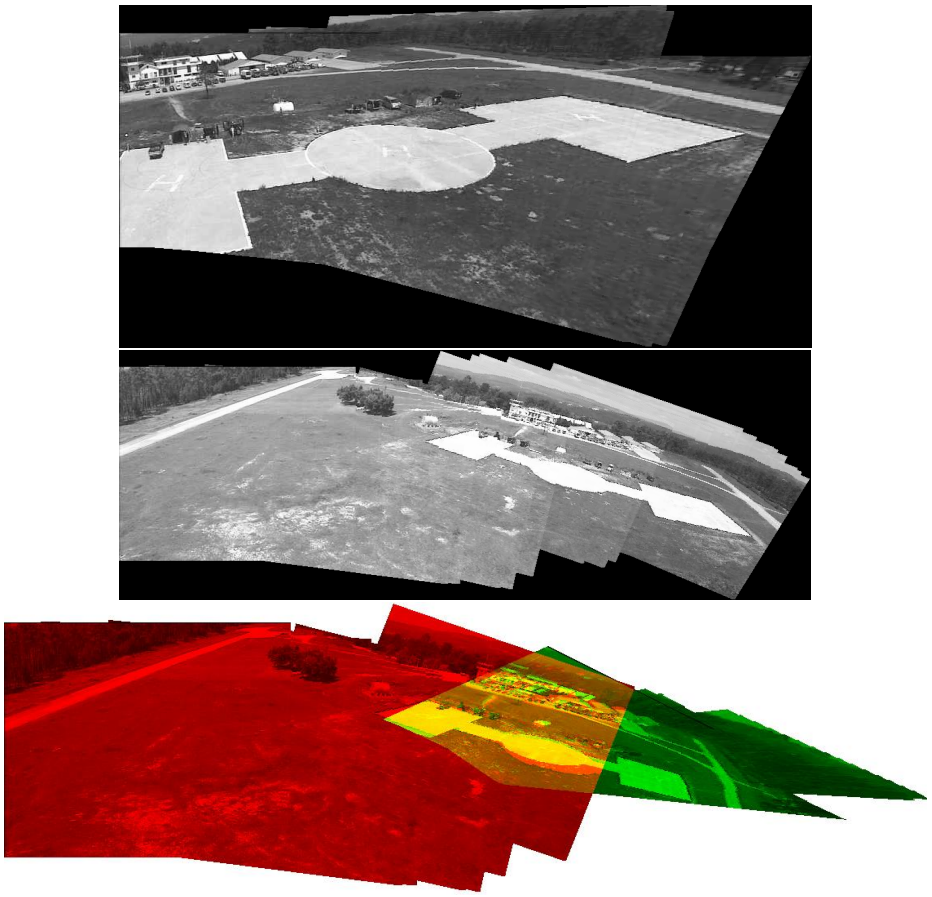

Figure 7.23: Mosaics built individually by Heliv and Marvin (top) and mixed mosaic built from the blob matches (bottom).

same UAV into two sequences. The images matched are gathered from quite different points of view. The same procedure is employed. Figure 7.25 shows the results.

If the initial relative pose of the different UAVs in the fleet is not known, this relative displacement can be used to incorporate constraints on these poses. Also, assuming that the position of the reference UAV in a global frame is known, it is possible to estimate the position of all the UAVs in the global frame.

In general exploration missions, the probability of having the same scene in the field of view of two or more UAVs is not very high. Also, as mentioned previously, it is necessary to have distinctive areas to get a reliable homography computation. If the scene is too homogeneous, the method would not work. Most of the images taken during the experiments, such as those in Figure 7.22, contained enough structure and the method worked satisfactorily. If the common area between the images is not large enough, it would be unlikely to obtain enough blob matches. In this case mosaics (Caballero et al., 2006) individually created by each UAV (see Fig. 7.23) can be used, at a cost of

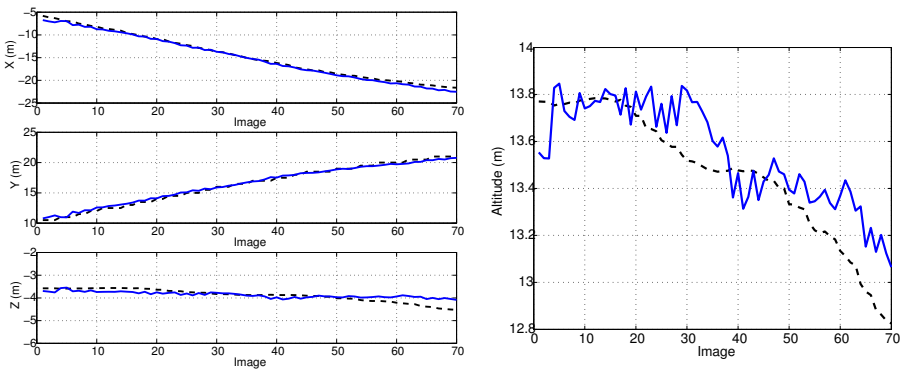

Figure 7.24: The homographies obtained from the blob matches can be used to estimate the relative pose of two UAVs. If one of the UAVs has a GPS receiver the other can obtain then its position in the same reference frame. In the figure, the relative pose of Marvin with respect to Heliv is combined with the local displacement estimation obtained by using the method of Section 7.2.2. Left: vision-based (solid) and GPS-based (dashed) estimations of relative displacement of Marvin with respect to Heliv. Right: distance to the ground using the procedure described (solid) and GPS measures (dashed).

memory storage (although it could be possible to store a blob representation of the mosaic, instead of the mosaic itself). The blob matching technique could be applied between an image and the mosaic built so-far . This would allow to obtain homographies even if the UAVs are not seeing the same scene at the same time at a cost of higher computational requirements.

7.5 Conclusions

The chapter has presented a visual odometer that can be used for motion estimation in UAVs. It employs matches between natural landmarks obtained from monocular images. Under a planar assumption, it is possible to obtain two different solutions for the relative displacement from the homography that relates consecutive images. More than two images can be used to disambiguate the final solution. The technique has been tested with data gathered during actual UAV flights, and it is quite accurate. Moreover, the same technique can be used to position the UAV relatively to objects.

Also, it is possible to use the same ideas to obtain the relative displacement between UAVs from matches gathered by their images. Point features are not adequate for matching images gathered by different cameras at disparate points of view. Researchers at the Linköping University have developed a blob-feature extraction algorithm that can be used for these tasks. Some initial results are presented. This can be used for cooperative perception, in order to anchor the data obtained by different UAVs to the same reference frame.

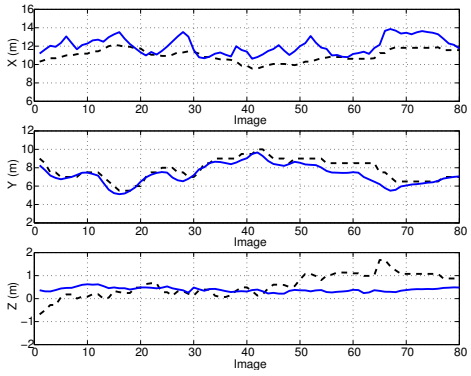

Figure 7.25: A second experiment. Vision-based (solid) and GPS-based (dashed) estimations of the displacement of one sequence expressed in the frame of the other by using the estimated relative transformation.

La experiencia es la única fuente de la verdad: sólo ella puede enseñarnos algo nuevo[...] Pero no es suficiente observar; es preciso usar de esas obsevaciones, y para ello es necesario generalizar.

Henri Poincaré
Ciencia e Hipótesis 1902

Chapter 8

A cooperative perception software system for multiple UAVs

The main part of the work of this thesis has been developed in the framework of the COMETS European Project. As stated in Chapter 1, the main objective of COMETS was to design and implement a distributed architecture for cooperative detection and monitoring using heterogeneous UAVs (Ollero et al., 2005).

Coordination and cooperation require a decisional architecture dealing with issues like task planning and allocation, coordination and supervision in a multi-robot fleet. This chapter summarizes the particular decisional architecture developed in COMETS and presents the perception architecture which provides the environment information required for the operation of the system. This perception architecture constitutes a software implementation of the ideas presented in the previous chapters of this thesis, and is also summarized in this chapter.

The project involved many people from many teams, and a strong emphasis was put on actual demonstration of the developed techniques. The chapter describes the experiments from which many of the results presented in this thesis were obtained. The vehicles involved, the actual conditions and system tested and some additional results obtained online during the experiments are presented as part of the chapter.

8.1 The decisional architecture in COMETS

In Chapter 2, the robot team involved in COMETS was classified in terms of the taxonomy by Dudek et al. (2002). One of the characteristics of the fleet is that the UAVs are heterogeneous. Actually, the multi-UAV architecture should integrate physically heterogeneous robots (helicopters and blimps), but it should also tackle with heterogeneity in terms of the decisional capabilities of the robots: some UAVs are directly controlled by an operator, some others are only endowed with *operational autonomy*, their tasks being planned and monitored by the control centre, whereas others may have *decisional autonomy* capacities, and hence can achieve given missions by themselves. Moreover, depending on the situation, an operator or the central station should be able to take control over

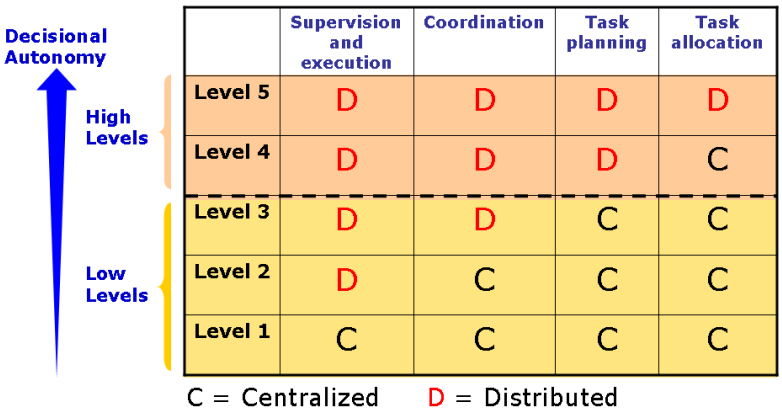

Figure 8.1: Decisional levels considered within COMETS. The higher the level, the more decisional capabilities are on board the UAV. Adapted from (Gancet, 2005)

UAVs endowed with decisional capabilities.

COMETS hence requires an architecture integrating both central decision making and distributed decision capabilities; moreover, this architecture should provide the possibility to dynamically configure the decisional scheme, depending on the available decisional capabilities of the robots and on the operational context. The design of this architecture is the main topic of the thesis of Jérémi Gancet at LAAS (Gancet, 2005; Gancet et al., 2005), and here the main aspects are summarized.

In a multi-robot system, decisional autonomy mainly considers the following issues: Supervision and execution of tasks, coordination, task planning and task allocation. These decisional components can be implemented according to different configurations: they can be gathered within the control centre (or Central Decisional Node, CDN), or be partially (or even totally) distributed among the UAVs. Five levels of decisional autonomy are defined within the COMETS architecture, according to whether these features are distributed or centralized (Gancet and Lacroix, 2004; Ollero et al., 2005) (see Fig. 8.1):

Level 1 No autonomy onboard the robot. The robot is only able to directly execute elementary tasks requested by the control centre (it is assumed that the UAVs at least are able to perform these atomic tasks, listed in Table 8.1).

Level 2 Executive capabilities (operational autonomy). The robot is able to manage partially ordered sequences of elementary tasks, and to return the execution status of the tasks.

Level 3 Same as level 2, plus simple interacting capabilities. The robot may manage online simple

Table 8.1: Atomic tasks performed by the UAVs

Atomic task	Description
TAKE_OFF	The UAV should take-off and reach a given altitude.
LAND	The UAV should land at a given position.
GO_TO	The UAV should move to a given destination waypoint.
TAKE_SHOT	The UAV should carry out a perception action (from gathering images to detection, depending on the parameters.)
WAIT	The UAV should stay (hovering or flying circles) in a secure mode.

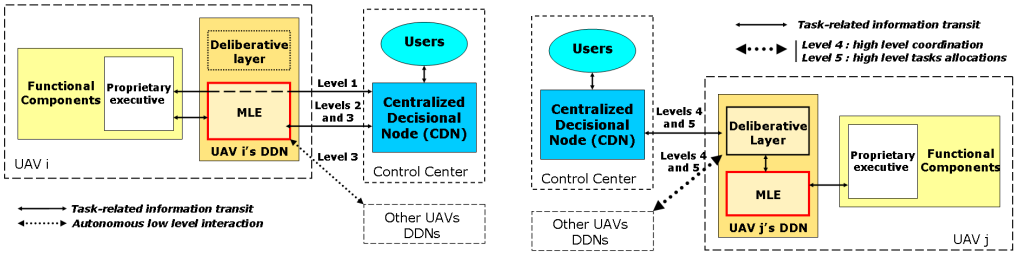

Figure 8.2: Left: Low levels. The MLE manages task sequence execution and even simple interactions with other UAVs (synchronization of tasks) up to level 3. However, task planning and allocation are carried out by the central station. Right: High levels. Task planning, allocation and high level plan coordination are completely distributed. Adapted from (Gancet, 2005).

interactions (synchronizations) directly with other robots endowed with at least the same level of decisional autonomy

Level 4 Deliberative capabilities. High level task requests are managed (involving autonomous task planning/scheduling), and the multi-robot task coordination is autonomously ensured in a distributed way among robots endowed with at least the same level of autonomy.

Level 5 Same as level 4, plus task reallocation capabilities. The UAV may opportunistically reallocate tasks and accept new tasks from other UAVs of the system also endowed with this level of autonomy.

This taxonomy is characterized by a large gap between levels 3 and 4: up to level 3, the control centre is expected to ensure the global consistence of the system's activity ("low levels" of decisional autonomy); whereas levels 4 and 5 introduce the possibility to delegate coordination and mission refinement activities in a distributed way ("high levels" of decisional autonomy).

Regarding low autonomy levels, the decisional layer on board each UAV is restricted to an

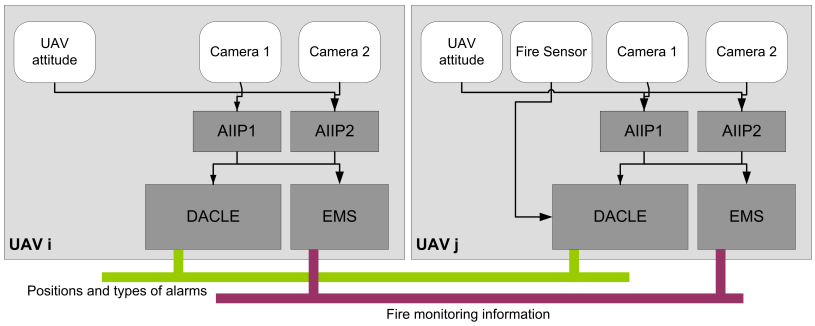

Figure 8.3: Decentralized architecture of the Perception System.

executive. This executive is denoted as the multi-level executive (MLE) as is common to all levels. At level 1, the MLE behaves as a transparent connecting point between the control centre and the UAV's functional components. However, the full MLE's features are required when considering upper levels: at levels 2 and 3, it manages tasks sequences execution, and at level 3 it enables simple interactions with other UAVs of the same level (see Fig. 8.2, left).

For higher levels, the MLE relies on the UAV's deliberative layer instead of the control centre, tackling higher autonomous decisional capabilities. The decisional layer deals with missions and task refinements, as well as coordination activities or task reallocation in a distributed fashion (see Fig. 8.2, right). More information about the particular capabilities of the decisional layer can be found in (Gancet, 2005; Gancet and Lacroix, 2004; Gancet et al., 2005)

8.2 The perception system architecture

Besides the decisional system, a Perception System (PS) was developed for the multi-robot team. This perception system provides the environment information required to accomplish the missions commanded to the decisional nodes of the fleet.

Following the arguments given in previous chapters, a decentralized approach is followed for the design of the perception architecture. A local perception subsystem is implemented for each UAV, which obtains local estimations from the sensors on board the particular UAV. This local subsystem consists itself of several modules (see Fig. 8.3):

- Application-Independent Image Processing (AIIP), which is the processing front-end, the module of the Perception System closest to the cameras. There is one AIIP module for each camera which obtains the required features for different applications.

- Detection/Alarm Confirmation, Localization and Evaluation Service (DACLE), whose objective is to perform fire detection/alarm confirmation and localization.

- Event Monitoring System (EMS). The EMS is in charge of the multi-UAV fire monitoring functionalities.

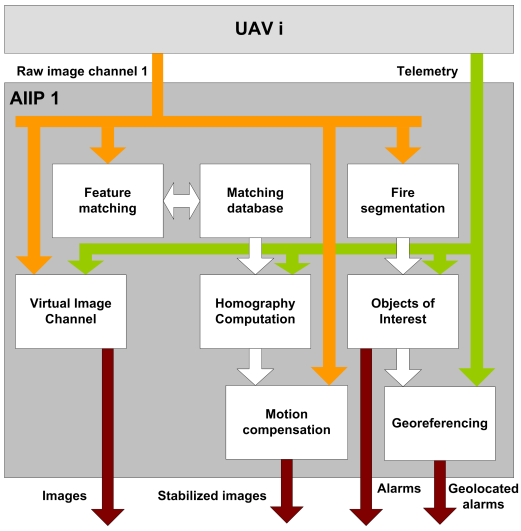

Figure 8.4: The AIIP module provides several services to other modules.

The information provided by the different perception modules can be accessed by the deliberative nodes (for instance, the Control Centre) for planning, replanning and other activities.

8.2.1 AIIP

The AIIP module allows applying a first processing step over the images, reducing its dimensionality (and hence, the bandwidth needed to transmit them). At the request of other modules, the AIIP module provides a set of functionalities (see Fig. 8.4), such as stabilization of image sequences (Section 6.7.1), fire segmentation (Appendix B) and geo-referencing (Section 3.5). Moreover, the AIIP tags each measurement with the required information in order to compute the corresponding likelihoods, like UAV and camera pose information and the detection capabilities of the segmentation algorithms.

Also, the AIIP acts as a virtual image channel, being able to modify the resolution and region of interest of the images, etc. The AIIP module is activated when the UAV receives a command for image gathering (the `TAKE_SHOT` task).

8.2.2 DACLE

DACLE is in charge of decentralized event detection and confirmation, localization and tracking. It maintains the information about objects of interest and all the associated uncertainties.

Figure 8.5, left, shows an scheme of the DACLE module. It incorporates the local measurements received from the different AIIP instances (segmented objects of interest and information to obtain their likelihoods) and other sensors to obtain a local belief state. Moreover, it shares its own belief with other DACLE modules corresponding to other UAVs, dealing with the issues of information fusion and removal of common information.

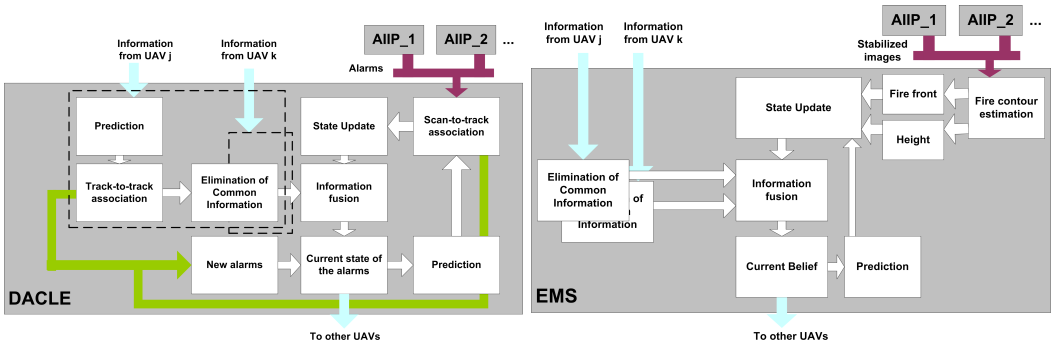

Figure 8.5: Left:Functionalities of the DACLE. Right: functionalities of EMS. Both systems are decentralized, and share information with the same systems on board the other UAVs.

Several implementations have been developed, considering different belief representations, following what is described in Chapters 3 (Information Filter), 4 (grid representations) and 5 (particle filters).

The DACLE module is activated whenever an UAV receives the task for detecting events (the `TAKE_SHOT` task with certain parameters indicating detection).

8.2.3 EMS

EMS is in charge of cooperative event monitoring. The particular application developed is fire monitoring, and therefore this module is in charge of obtaining and maintaining all the information regarding the evolution of the fire.

As input data, it uses stabilized image sequences provided by the AIIP modules, and employ the techniques described in Chapter 6 for estimating the evolution of the fire front and other parameters. Moreover, these estimations are shared with other UAVs for cooperative monitoring (Fig. 8.5, right).

The EMS module is activated also when a `TAKE_SHOT` task is received, in this case with a parameter indicating fire monitoring.

8.3 Communication

The distributed software architecture employs a custom communication system, called Black Board Communication System (BBCS), as communication layer for the different modules. The BBCS, developed by the Technical University of Berlin (Remuß and Musial, 2004; Remuß et al., 2004), is implemented via a distributed shared memory, the *blackboard* (BB).

A network node (NN) of the BBCS is a task, whether in a multi- or single-tasking environment. In the case of the COMETS system, these tasks are the processes implementing the decisional nodes and the different perception subsystems. From an abstract point of view, all NN using the BBCS, shares an area of memory (see Fig. 8.6).

The BBCS network follows a server-less peer-to-peer approach that is able to mimic various topologies since NNs are able to impart information between adjacent NNs. The BBCS can provide

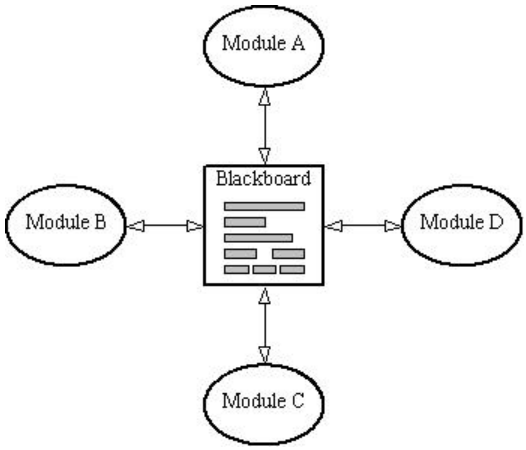

Figure 8.6: Abstract perspective of the communication system (Remuß and Musial, 2004).

reliable service using unreliable channels because it handles errors, such as data losses, multiple or out of order reception.

As said before, the BBCS is realized via a *distributed* shared memory, the blackboard. Each NN has a *local* copy of the BB portion it is accessing. The BB is organized in slots. Each slot is associated with a specific class of information and can hold a specific amount of data, for instance the current position of a UAV, the belief state about the position of an event or pictures and video streams.

The consistency of this shared memory is ensured by a real-time aware protocol. The BBCS API also offers a set of functions to deal with wireless communications and include functions robust to periods of degraded bandwidth, not infrequent in forest scenarios. Its high configuration capability allows implementing network communications with low delay using a simple software structure.

To be able to use standard network components, the BBCS was designed as an overlay network on top of existing transportation layers (UDP, TCP). It can be adapted to different kinds of operating systems and hardware platforms (ranging from PCs to microcontrollers), always offering the same services and interfaces. In the COMETS system, the network includes wired and wireless links to connect all systems; nodes are mostly usual PCs running Windows or Linux, embedded PCs, but also micro-controllers. This is one of the main differences of BBCS with respect to other approaches: BBCS is very tiny, and can run even in small micro-controllers, which are usually the computational payload for small UAVs. More details can be found in (Remuß and Musial, 2004; Remuß et al., 2004).

8.4 The COMETS robot team

The flying segment of COMETS is composed of three aerial vehicles.

Figure 8.7: The Marvin autonomous helicopter. Left: Marvin I. Right: Marvin Mark II.

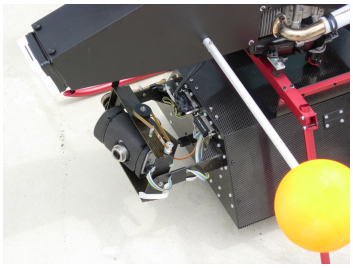

Figure 8.8: Marvin's digital camera mounted on the pan-and-tilt unit.

8.4.1 The autonomous helicopter MARVIN

Marvin (Remuß et al., 2002) is an autonomous helicopter developed by the Real-Time Systems and Robotics Group of the Technische Universität Berlin (TUB). Two different prototype versions of the helicopter were developed along the project[1]. Figure 8.7 shows both versions.

Although the airframe and electronic box are different, in both cases the helicopter is equipped with a strap-down IMU, a compass, an ultrasonic rangefinder, a precision DGPS and a microcontroller for autonomous flight. It also carries a high-resolution digital camera and a PC-104+ for image acquisition, fire and temperature sensor for fire detection and two redundant radio links for telemetry, commands and image transfer. An earlier state of the system has won the International Aerial Robotics Competition 2000 held by Association of Unmanned Vehicle Systems (AUVS).

As commented, the main exteroceptive sensors of Marvin are a fire sensor and a digital camera. The main component of the fire sensor is a photodiode set-up to limit its sensibility to the band of [185, 260] nm, normally associated to fires. The output of the sensor is a scalar value, proportional to the radiation energy received. Marvin also carries a Canon S40 digital photo camera and a pan and tilt unit that allows orientating the camera independently from the body of the vehicle (see Fig. 8.8). The unit has encoders that measure the pan and tilt angles.

[1] http://pdv.cs.tu-berlin.de/MARVIN/mark_ii_frameset_introduction.html

Figure 8.9: The Heliv teleoperated helicopter and its electronic box (the pan and tilt unit can be distinguished).

8.4.2 The teleoperated helicopter Heliv

The Heliv helicopter (see Fig. 8.9, left) is the result of the evolution of a conventional remotely piloted helicopter of the Spanish SME company HELIVISION, which has been transformed by the Robotics, Vision and Control Group at the University of Seville by adding sensing, perception, communication and control functions (Ollero et al., 2003).

Heliv carries a GPS receiver, an IMU and other sensors for attitude and position estimation. Moreover, Heliv is equipped with infrared and visual video cameras for perception purposes. Each video camera is connected to a video server which digitizes and sends the image streams using standard net protocols.

The infrared camera is a low-cost non-thermal OEM micro-camera in the far infrared band (7-14 microns). The visual camera is a low-weight color device with 320x240 pixel resolution. Heliv also carries a motorized pan and tilt unit to orientate the cameras (Fig. 8.9, right).

Heliv incorporates a teleoperation station that allows a pilot to be integrated into the COMETS decisional architecture. This teleoperation station receives atomic tasks from the Heliv MLE (its executive) and responds with the status of the tasks (started, running, ended, etc.). On the other hand, the teleoperation station guides the pilot to accomplish the given tasks through visual and audio interfaces.

8.4.3 The autonomous blimp Karma

Karma (see Fig. 8.10) is a $18m^3$ electrically propelled airship developed at LAAS (Laboratoire d'Architecture et d'Analyse des Systèmes) (Hygounenc et al., 2004).

The blimp carries an electronic board for the control of the blimp and the acquisition of data. It employs an IMU, GPS and other sensors for state estimation and control. Also, Karma carries a stereo bench with two visual cameras in order to generate depth maps [2].

[2] http://www.laas.fr/~simon/eden/gallery/karma.php

Figure 8.10: The airship Karma.

8.5 The scenarios

8.5.1 General mission

As a guide for the development of the project, a general mission was considered. This general mission was designed in order to address all important issues in the project, like coordination, cooperative perception, etc., while at the same time being realistic and general enough. For instance, this mission could be part of the activities of a fire brigade. The typical script of a mission of this kind is the following:

1. As the fire risk is high, the chief of a fire brigade at Lousã (Portugal) selects two areas Z1 and Z2. UAVs C1 and C2 will be sent (respectively) toward Z1 and Z2, where new fires are likely to occur. C1 and C2 should perform detection during one hour, before coming back if no fire is detected.

2. A "possible fire" alarm is detected by C1 and signalled at position P. A third UAV C3 is required to confirm the possible alarm. The Control Centre (CC) plans and sends the mission for alarm confirmation over P to C3. In the same time, current C1's detection mission is interrupted: the CC re-plans C1's activity so that it stays around P, waiting for the confirmation of another UAV (e.g. C3). C3 takes-off, and reaches P after a few minutes.

3. C1 and C3 combine their local perceptions. The alarm is confirmed: the possible alarm appears to be a real alarm. In order to efficiently monitor the detected fire, firemen decide to request a cooperative monitoring with 3 UAVs: C1, C3 and B1. They specify mission updates to the CC, and the CC transmits the new missions to each UAV. C2 is required to interrupt its detection activity, and to come back and land. B1 takes off and moves toward P. Blimp B1 is required to fly high, around P's location, to get a general overview of the situation. C1 and C3 are expected to coordinate closely their activities, in order to get close, accurate perceptions of the fire from different perception angles. C1 is required to perform frontal view perception while C3 is required to perform lateral view perception.

4. During the re-planning of the COMETS fleet activity (aiming at fire monitoring), the firemen call the Lousa Fire Station to request some more fire fighting means: 3 trucks, and 10 to 12 other firemen. The monitoring activities of C1, C3 and B1 will help the fire brigades in the fire fighting activities.

5. The fire seems to be extinguished. A second blimp B2 is commanded to perform a terrain mapping task. The other three UAVs are commanded to come back and land.

6. End of the mission. Each UAV should return to its base when the operator decides to end either the detection, monitoring and mapping mission.

8.5.2 Activities involved

The previous mission requires that the UAVs are able to perform, among others, the following activities:

Alarm detection and localization

The global purpose is to detect an alarm of a given predefined type in the mission area and to obtain an initial estimation of the position of this alarm. Hence, an UAV or a group of UAVs is commanded to survey a given area. The task decomposition of this mission is carried out by assigning a given path along the selected operation zone to each UAV involved. The main task for the UAVs is to follow the path and capture images of the near environment with a pre-selected resolution and rate, and apply perception techniques for detection and localization of the potential alarms. If several UAVs are considered, the planning activities will coordinate the paths assigned to the vehicles, in order to cover the area completely.

From the perception point of view, each UAV surveys a different area, and thus there is no explicit cooperation, although the UAVs can still share information if they are within communication range.

Alarm confirmation

If an alarm is generated in the previous activity, it could be necessary to confirm it, due for instance to a big uncertainty in the nature of the alarm (in order to discard potential fire alarms).

This activity requires to re-schedule the available means, sending additional vehicles to the initial estimated position of the alarm and combining the information of the different UAVs to update the estimation of the type of the alarms, to geo-locate the alarm source precisely and to obtain additional knowledge of the alarm.

The main task for the UAVs is to follow the planned paths around the location of the possible fire alarm, to capture images or other sensorial data of the near environment with a pre-selected resolution and rate, and to combine these local data with the estimations of the companion UAVs to confirm or discard the potential alarm and to determine the position of the alarm with high precision.

Figure 8.11: The field experiments site. In the left photograph, the landing tracks for airplanes can be distinguished. The right picture was obtained by Marvin and shows on the left the landing zone for the UAVs.

Event monitoring

The global purpose of this mission scenario is to monitor the time-spatial evolution of an event of a given predefined type. Ideally, this activity requires the cooperation of several UAVs that will observe the event from different and complementary points of view. Furthermore, in some cases both panoramic and detail views could be needed for an appropriate visualisation of the scene.

The main tasks for the UAVs is to follow planned paths to the corresponding vantage points and to perceive *coordinately* the event, fusing the information from these different points of view to obtain a detailed model of the event.

8.6 Description of the COMETS experiments

In order to test the previous scenarios, a set of experiments involving several UAVs were scheduled along the project. This section describes those experiments, the actual missions performed and the results obtained.

8.6.1 The field

The COMETS project was demonstrated during three annual experiments carried out at the Lousã airfield, close to Coimbra, in Portugal. Figure 8.11 shows some photos of this airfield. The airfield is employed as landing platform for planes involved in fire fighting activities. Moreover, the place is a training site for fire brigades, and thus it was possible to perform small controlled fires with the help of firemen.

8.6.2 The actual demonstration

During the experiments, a reduced implementation of the scenario depicted in Section 8.5.1 was actually demonstrated. The following paragraphs present a summary of a typical execution of this

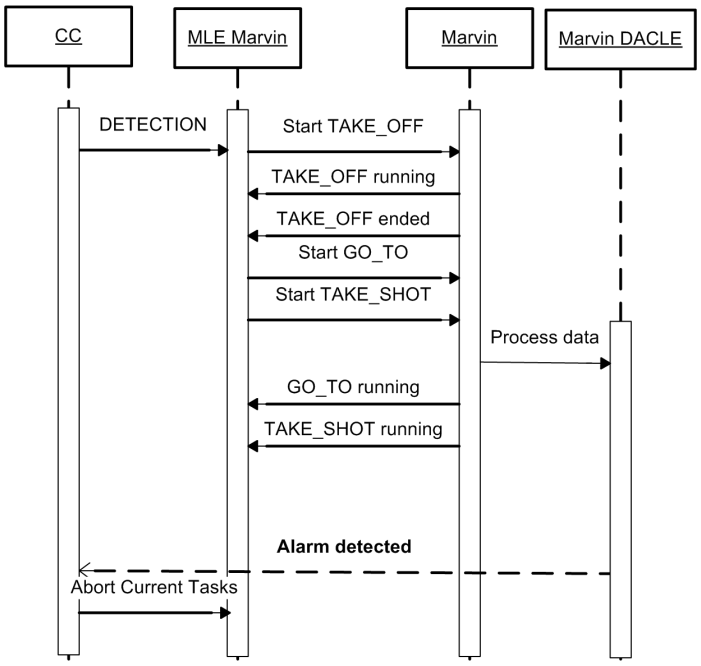

Figure 8.12: Exchange of tasks for the detection phase. A set of atomic tasks is sent to Marvin to cover an area and detect alarms.

scenario.

A fire detection mission is planned for Marvin (see Fig. 8.12). The corresponding sequence of tasks is sent to Marvin's Multi-Level Executive, which itself sends the atomic tasks to Marvin. Marvin processes its given tasks one by one . After taking off, Marvin follows the waypoints that it has received (`GO_TO` task). At the same time, DACLE processes the data gathered by its fire sensor and the images from its camera (`TAKE_SHOT` task). As a result of the processing, Marvin will eventually generate a potential fire alarm at a given location (depending on the test, sometimes more than one alarm is generated). The alarms are sent back to the CC.

After reception of a possible fire alarm/s, the operators at the CC define a mission for Heliv (see Fig. 8.13), consisting of a path to the potential alarm and a confirmation task (`TAKE_SHOT` task with different parameters). If several alarms are detected, the operators select which one to confirm/discard. The MLE of Heliv transmits to the pilot the corresponding tasks. The pilot achieves its tasks using the interfaces of the teleoperation station. At the same time, the current mission of Marvin is aborted and Marvin is commanded to fly to a hovering position farther from the fire.

The IR images of Heliv are processed and integrated by its local DACLE module. This local information is fused with the received information from Marvin. The DACLE system updates the probabilities and position of the alarms, discarding the false alarms and confirming the actual ones. All these data are also received by the CC. When the probability of fire reaches a predefined threshold, it is considered as confirmed, and a new replanning phase begins. Current missions of

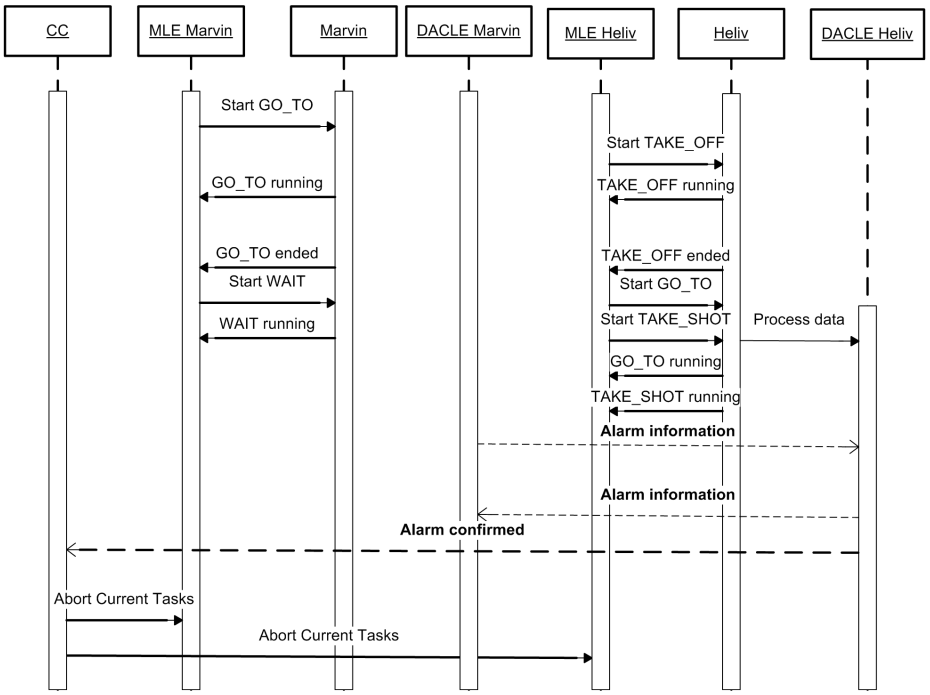

Figure 8.13: Scheme of the exchange of tasks for the confirmation phase. Heliv is commanded to go to a close position of the detected alarm and to take and process images to confirm or discard the alarm. It uses the information provided by Marvin. Meanwhile, Marvin is commanded to wait at a safe place.

Marvin and Heliv are aborted.

The operator generates a monitoring mission for Marvin and Heliv, involving synchronization tasks to take pictures of the event from the correct viewpoints at the same time. The MLEs of the vehicles are in charge of managing the synchronization signals in a decentralized manner (Gancet et al., 2005) (see Fig. 8.14). When the synchronization is correctly achieved, Marvin and Heliv begin to obtain pictures of the fire simultaneously and for a given time. Also, Karma is commanded to take images of all the area from a high vantage point.

The pictures are processed by the AIIP module of each UAV. This system stabilizes the images received and sends them to the EMS. The EMS processes the images and obtains an estimation of the position of the fire front. These results are shared between Marvin and Heliv, and sent to the CC.

After the time for taking pictures has expired, Marvin and Heliv are commanded to return home and land. At the same time, a mapping mission is generated for Karma (Lemaire et al., 2005). Afterwards, the mission is terminated.

The scenario execution shows the full COMETS system working in a real situation. A fleet of heterogeneous vehicles is controlled to achieve a complex mission. The mission involves real-time coordination of the vehicles.

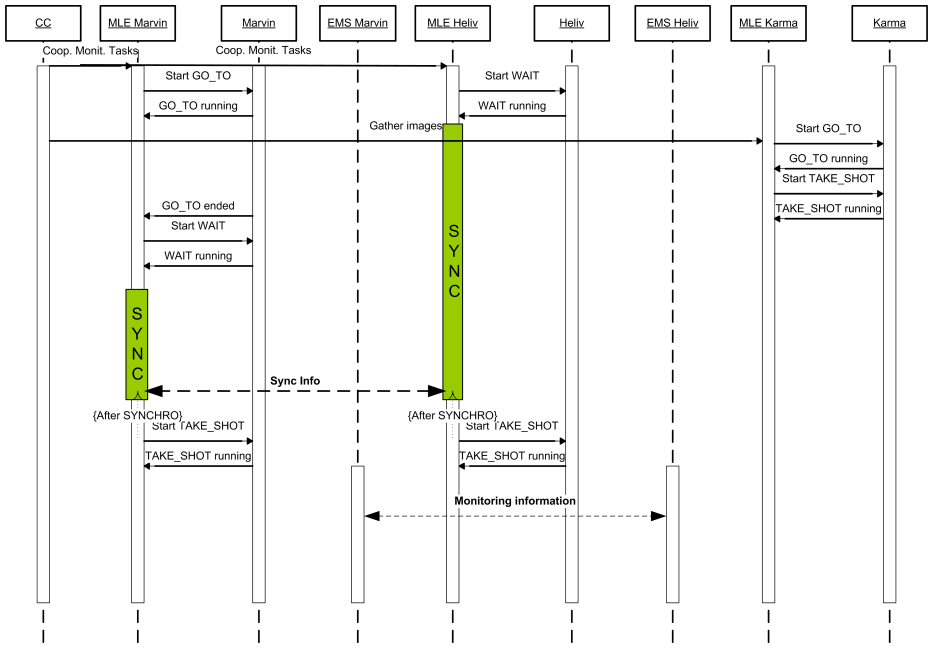

Figure 8.14: Exchange of tasks for the monitoring phase. The UAVs can synchronize their tasks, and therefore they can begin the monitoring of the fire from different points of view at the same time. The synchronization signals are employed as preconditions for the tasks to be synchronized.

8.6.3 Details of the tested system

During the experiments, the COMETS multi-UAV system was tested online. The particular implementation of the system during the experiments is described here.

The decisional architecture is tested in Level 3, that is, task planning and allocation occur in a centralized node (the CC), but task supervision and execution and task synchronization are completely distributed.

Regarding the PS, Marvin and Heliv have not enough processing power onboard to perform image analysis, and therefore their images are relayed to the ground, where the PS processes are running. Besides, although Karma carries more processing power on board, its images are also relayed. The physical layer employed is wireless ethernet. Nevertheless, the software architecture is decentralized, as depicted above, with one process for each vehicle and the distribution of modules described.

For the detection and confirmation phases, the main objective is to determine the precise position of the fire alarms, while discarding potential false alarms. The state of each alarm in this case is defined as:

$$\mathbf{X}_t = \begin{pmatrix} \mathbf{P}_t \\ \boldsymbol{\theta} \end{pmatrix} \tag{8.1}$$

where $\boldsymbol{\theta} \in \boldsymbol{\Gamma} = \{fire,\ no\ fire\}$ and $\mathbf{P}$ is the position of the alarm.

Figure 8.15: During the demonstration, small controlled fires were performed.

If the image objects obtained by the processing algorithms are associated to the current alarms, then it is possible to estimate separately (that is, employing a different filter) the position of the alarm and the classification $\boldsymbol{\theta}$. The position on the image plane is used to update the information about the position of the alarm, while the classification of the alarm is updated through the capabilities of the segmentation algorithms.

In the experiments, for the estimation of the *position*, the actual implementation of the algorithms for DACLE was based on the Information Filter, described in Chapter 3. Each UAV maintains a Gaussian distribution for the position of every alarm. Moreover, a Bernouilli variable is employed to represent the probability of being fire for each alarm, and therefore, the belief in this case is updated by using the decentralized binary Algorithm 4.3 of Chapter 4.

However, for Marvin, a hybrid approach is employed. The information from its fire sensor cannot be integrated directly into the Information Filter. Therefore, an evidence grid (Chapter 4) was used to incorporate the new data obtained from this sensor. From this grid, when a high likelihood region is detected, a Gaussian approximation was made by using equation (4.20). The mean position of the region and its covariances are incorporated within the Information Filter framework for determining the precise position of the fire alarms. The mean probability of the grid for that region is used as the initial estimation of the probability for $\boldsymbol{\theta} = fire$.

During the monitoring phase, the EMS is activated. The stabilization, fire segmentation and fire contour extraction algorithms described in Chapter 6 were applied on line to the images provided by the UAVs. The local estimations are shared and combined.

In order to clarify these aspects, the next section describes the results obtained in one of the experiments.

8.6.4 Perception System results

The results from the perception processes during one of the experiments are described here. The section focuses on results that were obtained online during the execution of the experiments. Besides, these data and data from other experiments have been also used for validation of other techniques presented in other chapters of the thesis.

Figure 8.16: Marvin employs a grid representation to integrate the data gathered by its fire sensor. The red square indicates the actual position of the fire. High probability regions are extracted and approximated by Gaussian distributions. Due to false alarms, other high probability regions are also detected.

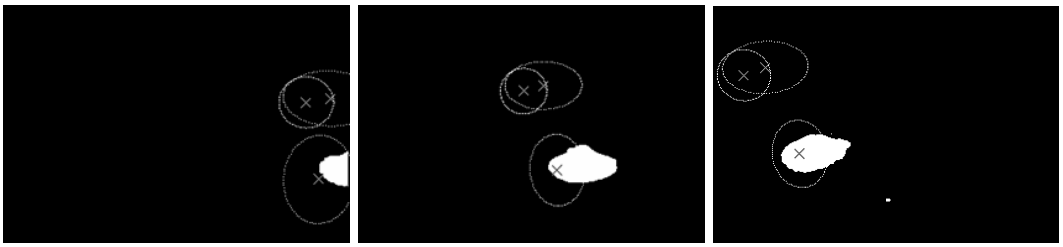

Figure 8.17: Heliv receives the information of the alarms from Marvin and updates its belief by using its infrared camera. This figure shows the estimated position and uncertainty of the alarms detected by Marvin projected on the Heliv's infrared camera.

In the experiments, small controlled fires were set by firemen and people from the project. Figure 8.15 shows a photograph of one of these fires. The position of the fire is recorded by using GPS receivers for validation purposes.

As commented above, during the first phase, Marvin is estimating the position of the potential fire alarms. For this, its DACLE process is employing a grid representation, and the procedures of Chapter 4 are used. Figure 8.16 shows the status of the grid in two instants of one particular experiment. After a while, several regions of high probability are detected. Each region is converted into a Gaussian representation for the position $\mathbf{P}_t$ and then incorporated into the Information Filter version of DACLE as a new alarm. Also, the mean probability of the cells of each region is used to estimate the probability that $\boldsymbol{\theta} = fire$ for that object.

When Heliv is sent to confirm the alarm, it receives estimations of the potential alarms from Marvin, in the form of an information vector and matrix for their positions and the log-odds l of being fire. Heliv then fuses the Marvin estimation with its local estimation obtained with its infrared camera. The images are segmented into potential fire objects. If one of these objects is associated to an alarm, its position is used to reduce the uncertainty on the estimated position of the alarm.

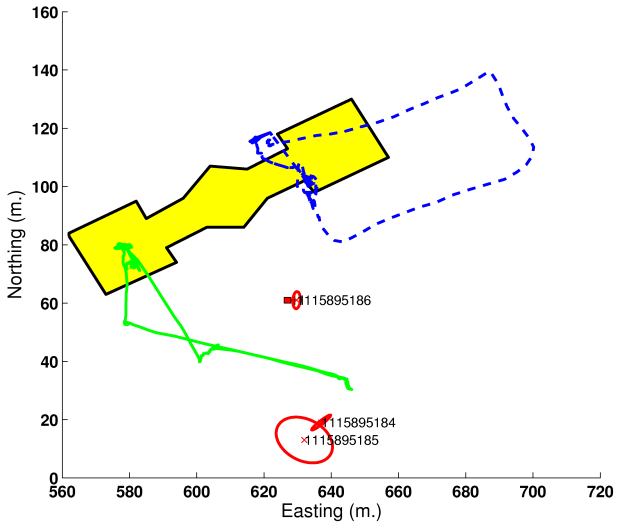

Figure 8.18: Final complete trajectories of Marvin (solid) and Heliv (dashed). The ellipses represent the final estimated position of the different alarms. The square indicates the actual fire position. In the experiment, besides the correct alarm, two false alarms are generated. The scheme also shows the landing platform from where the UAVs take off.

Moreover, the probability $p(\boldsymbol{\theta} = fire)$ is increased for this alarm.

Moreover, the scheme allows using negative information. If one alarm is within the field of view of the Heliv's camera, but it is not associated to a segmented object, then its probability $p(\boldsymbol{\theta} = fire)$ is decreased, although nothing can be said about its position. Figure 8.17 shows the potential alarms estimated by Marvin projected on the images of Heliv. It can be seen how the actual fire alarm is associated to the segmented fire object, while the false alarms do not correspond to fire regions.

Figure 8.18 shows the final trajectories of both UAVs, and the final estimated position of the alarms. The estimated position of the actual fire alarms is within 1 meter of the correct position. The evolution of the estimation on the position error on the DACLE of Marvin is shown in Fig. 8.19. Also, the probability of the alarms is shown. It can be seen how the false alarms are discarded.

After the fire detection mission, the fire monitoring mission begins. During this, Marvin and Heliv are commanded to obtain views of the fire from different vantage points, and each local EMS applies the procedures described in Chapter 6 to stabilize their images, segment the fire fronts and estimate their position. In these experiments, the fire is not big enough to appreciate an evolution on the fire front, but Fig. 8.20 shows how the shape of the fire is correctly estimated. Karma is also sent to obtain images about the zone, which are received by the processing units and stored.

Other results

During the last year session, more that 20 fire detection and confirmation missions where performed, involving the three UAVs. In those missions, the fire alarm was detected in more than the 95% of

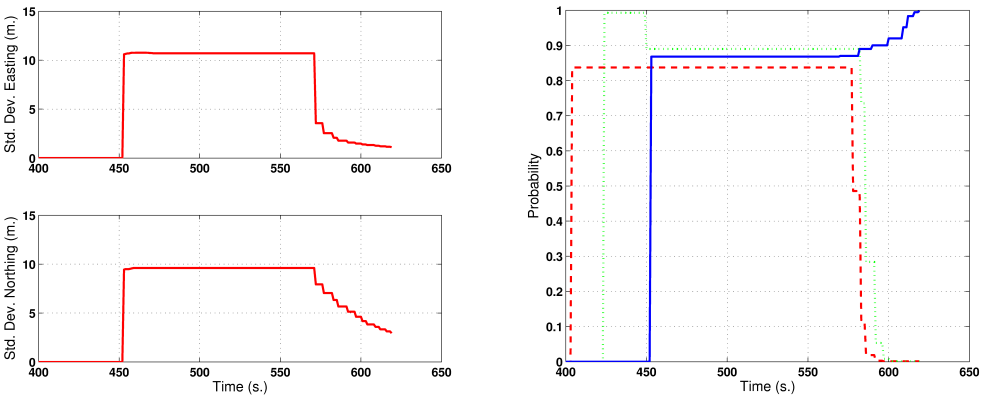

Figure 8.19: Left: estimated standard deviation on the error. First, when Marvin detects the alarm the uncertainties are high due to the poor localization capabilities of the fire sensor. When Heliv fuses the information with its camera, the uncertainty decreases. Moreover, the probability of being an actual fire alarm is updated accordingly.

the cases, and more than the 80% of the false alarms were correctly discarded. The precision of the location of the fire alarm is below 1 meter.

8.7 Conclusions

The objective of this last chapter before the final conclusions of the thesis was three-fold. First of all, to complete the framework in which the previous ideas and algorithms have been developed. It has been presented how the decisional architecture of COMETS combined with the perception techniques described in this thesis allows an UAV fleet to perform cooperative perception missions.

Second, to balance the pure scientific and applied aspects of the thesis. The results presented show the feasibility of the ideas. It should be remarked that the results presented in this chapter were all of them obtained online during the experiments.

Third, to illustrate the capabilities of UAVs in the field of environment surveillance and fire fighting in particular. Even though the presented tests cannot be considered as completely operational, they are a good approximation and are relevant enough to show the potential of these robots and techniques.

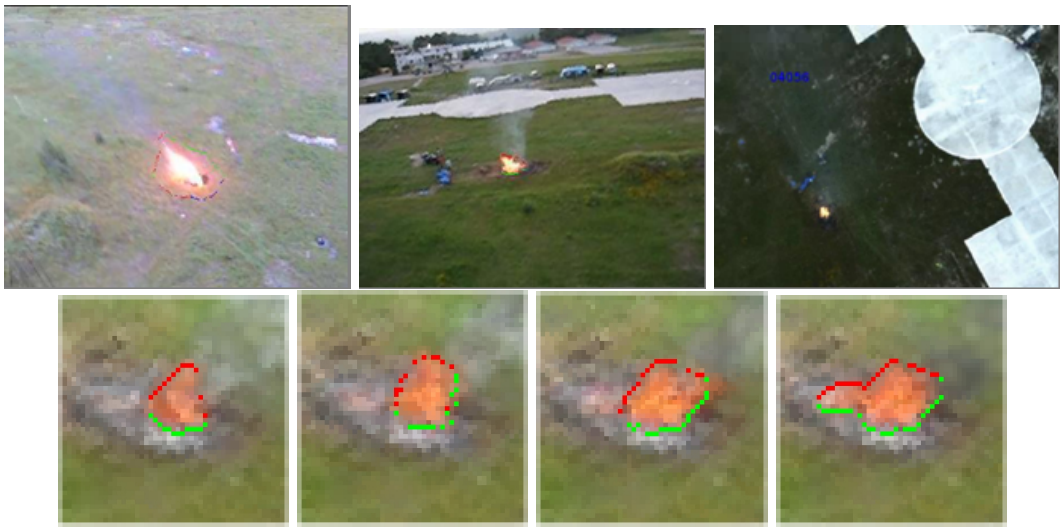

Figure 8.20: Top: left, image from Heliv, after stabilization and feature extraction; middle, image from Marvin after stabilization and feature extraction; right: image from Karma. Bottom: details of the extracted contours. Green: fire front. Red: top of the flames.

...All those moments will be lost in time, like tears in the rain.

Roy Batty
Blade Runner

Chapter 9

Conclusions and perspectives

The objectives of this thesis were to provide contributions to the field of multi-robot systems, concretely for cooperative perception in systems of multiple UAVs. The application of the ideas to an actual system and its demonstration in a real application have been an important guideline for the development of the thesis.

In this chapter the main contributions are summarized, and their pros and drawbacks discussed. The chapter ends presenting potential future developments.

9.1 Revisiting the main contributions

9.1.1 Summary of contributions

The main contributions of this thesis can be classified in three main axes:

- Multi-robot cooperative perception. Bayesian approaches are analyzed for the design of an architecture for multi-robot perception. The framework allows for decentralized information fusion. Different implementations considering different representations of the belief states are proposed. Chapters 3, 4 and 5 analyze parametric (Information filters), and non-parametric (grids and particle filters) representations. The architecture and the different approaches are applied to multi-UAV decentralized detection, precise localization and tracking of events. The different schemes have been tested. Simulation results are used to illustrate the techniques. Vision is used as main source of information, but other sensors are considered as well.

- UAVs for environment monitoring. The previous ideas are applied to the actual problem of fire detection and localization by a fleet of UAVs. Results from actual field experiments involving 3 UAVs, equipped with infrared and visual cameras and fire detectors, have been provided. Moreover, Chapter 6 presents a system for automatic forest fire monitoring using vision. This system can consider information provided by multiple UAVs for forest fire monitoring. Results obtained from field experiments with controlled fires are presented.

- Vision-based UAV navigation. Chapter 7 presents results on a potential visual odometer for

UAV navigation, and an initial extension for multi-UAV relative position estimation. The system does not rely on artificial landmarks. Results from data gathered in different experiments are presented.

Some of the results presented along this thesis have been obtained online during the general experiments of the projects COMETS and SPREAD. Those experiments consider close-to operational scenarios, and, in the opinion of the author, they constitute one of the main contributions of the thesis, as they illustrate the applicability of the techniques and ideas presented in the real world.

9.1.2 Detailed discussion

Decentralized multi-robot perception

One of the main motivations of the ideas of the thesis is that fleets of robots can accomplish some tasks that one robot alone is not able to do; additionally, they can accomplish the tasks better than a robot alone. As commented in Chapter 1, UAVs are a very interesting tool for information gathering tasks, and thus the thesis has focused on techniques for cooperative perception.

A framework for multi-robot perception is analyzed in Chapter 2, based on probability theory. Probabilistic algorithms for robot perception, action and learning are widely used in the robotics community. Acting in outdoor and dynamic environments requires handling uncertainty, and Bayesian inference is a well founded tool that allows integrating noisy measurements and uncertain actions to update the knowledge of the world.

Other representations and frameworks for uncertain reasoning are possible, but, in the last years, probability theory has provided robust solutions to complex problems, like SLAM (Thrun et al., 2005). As shown, the same Bayesian framework can be extended for information fusion when there are several sources of information (as is the case with multi-robot systems); although not further investigated in this thesis, mixing this with utility theory can provide a framework not only for information fusion, but also for developing cooperative information gathering actions, and for planning under uncertainty.

Of course, other approaches have proven to be useful for information fusion. Thus, some authors employ Dempster-Shafer evidence theory for multi-robot perception, as for instance Yanli et al. (2005). However, the extension of this approach for decentralized fusion or the development of cooperative information gathering actions is not clear.

In the thesis it has been argued in favor of decentralized schemes for information sharing, in which only local interactions are allowed, seeking for scalability. However, in small fleets of robots, centralized solutions could be sufficient for accomplishing some information gathering tasks, and centralized schemes can theoretically provide better results. In fact, other topologies for the information flow and more complex information exchange protocols could provide higher benefits from the fusion, at the cost of decreasing the flexibility. Nevertheless, if the applications are to be extended to fleets of tenths of robots in outdoor applications, then decentralized or at least partially decentralized schemes are a must-have. Moreover, from the experience during the field experiments it has been seen that, even in the case of small fleets, communication failures among UAVs and

with the ground stations are very likely to happen, if non special wireless communication devices are employed. Decentralized schemes are much more robust under these circumstances, as they can withstand temporal communication losses, new robots can be added or removed from the fleet with not interference, and evidence can be accumulated locally and transmitted depending on the circumstances.

As discussed, decentralized information fusion raises the problem of rumor propagation, which can lead to non-consistent estimations (due to the lost of independence in the sources). This has been pointed out mainly by Grime and Durrant-Whyte (1994), Sukkarieh et al. (2003) and other publications of these authors, although this problem is usually not considered in the robotics community. Fixed topologies for communication can be used to overcome this problem, like tree-like structures, but they are too rigid for fleets of mobile robots, in which the communication links will change. In general topologies, this is a problem that will occur, and that has to be considered.

Efficient communication calls for parametric representations of the local belief states. And therefore, Chapter 3 shows how the Information Filter uses a quite efficient representation that also has very interesting analytical properties. Moreover, Gaussian distributions provide analytical tools (the Covariance Intersection algorithm) for combining information conservatively, even in presence of unknown correlations. These fusion schemes are suboptimal in the sense that the resultant variance obtained is always greater or equal than that obtained in a pure centralized case, but they guarantee consistency of the estimation. Also, for Markov processes the information matrix is sparse, as shown in Section 3.1.2. In fact, these filters have been applied to problems in which the state is high dimensional (Thrun et al., 2004). However, the efficiency comes at the cost of flexibility, as they can actually be applied only in certain situations. Linearizations and other schemes, as the Unscented Transformation, allow extending the applicability of these representations, but care should be taken in order to avoid divergences.

There are several contributions on the use of Information Filters for multi-robot perception, being (Nettleton et al., 2003; Sukkarieh et al., 2003) the most related to this work. However, exploiting the sparse structure of the Information matrix for tracking mobile objects, exchanging full trajectories, is believed to be novel. Moreover, many information fusion results do not consider the uncertainties on the pose of the sensors, which should be definitely included and are considered in the proposed technique. Moreover, the Information Filter considered in Chapter 3 has been tested online during the experiments described in Chapter 8.

Nevertheless, there are aspects that are better described with discrete representations (as classification and identity of objects). These representations are considered in Chapter 4, where decentralized information fusion employing certainty grids is analyzed. Certainty grids have been used for a long time in the community for tasks as mapping. They are a nice representation for static environments, and are more general than the parametric Gaussian representation.

There are approaches for multi-robot exploration using evidence grids, but less work can be identified on decentralized estimation using certainty grids. The approach presented here considers uncertainties on the sensor pose in the estimation phase and allows mixing data from heterogeneous sensors, ranging from infrared images to simple fire detectors. Also, it is discussed the possibility of

applying other more conservative combination rules for these grids. Nevertheless, binary certainty grids lack of information about the support for the estimated probabilities. Beta distributions can be used to overcome this problem. Bertucelli and How (2006) have considered the use of Beta distributions in certainty grids, but focused on the control problem for optimal exploration. The decentralized scheme described, employing Beta-based evidence grids, is considered unpublished. Moreover, this scheme allows to derive conservative fusion rules for the fusion of evidence grids. These rules are also new.

However, the certainty grid representations are less efficient for communication. Moreover, they scale badly for higher dimensional problems, and thus for dynamic environments or 3D maps the storage and bandwidth requirements make other approaches preferable. Chapter 5 presents a quite flexible tool for perception, particle filters. They allow considering estimation under non-linear prediction and measurement models. Moreover, the implementation of particle filters is simple and can be (up to some extent) parallelized; and even its computational complexity (the number of particles) can be adaptively controlled. In the last years there has been contributions to the development of schemes for decentralized estimation employing particles (Rosencrantz et al., 2003; Ihler et al., 2004). These proposals apply fusion rules directly on the particle representation, as the approach presented here. However, for decentralized data fusion and data association, some operations require the recovering of some functional representation of the belief or to recur to approximations, which is not considered by those authors. This thesis employs a similar approach to (Ong et al., 2005b), employing Gaussian Mixture Models (GMMs) for recovering a functional approximation of the particle representation. However, in this case, the recovered model (which complexity can be controlled) is only used to re-weight the particles received, and therefore there is no need to obtain a new set of samples, as the received ones are directly included within the local estimation filter. The resultant filter is employed for multi-UAV localization of objects.

The main drawback is the computational complexity of obtaining the functional representation. A much simpler schema based on Consensus Theory is also analyzed for particle fusion. The fusion results are less efficient, but it can be also used for fusion, and the required operations are simpler.

Nevertheless, the use of mixtures for belief representation is a very interesting option. Mixtures are representations that try to mix the best of both worlds, flexibility and efficiency. Recently, GMMs have been employed for probabilistic perception. For instance, GMMs are being used in vision applications to represent the likelihood and priors associated to bearings-only measurements (Solà et al., 2005; Lemaire et al., 2005). In the author's opinion, these models are also quite suitable for a set of tasks in decentralized perception. Firstly, they can cope, up to some extent, with problems in which the measurement and prediction functions are non-linear. Also, the use of a mixture of parametric models reduces the amount of bandwidth required to communicate the beliefs. It seems that a combination of local particle filters for initial estimation in highly non-linear situations, followed by mixture model-based filters for information fusion and communication can be a flexible and efficient solution for multi-robot estimation.

Unmanned Aerial Vehicles for environmental monitoring

It is the opinion of the author that the utility of unmanned aerial vehicles for environment monitoring has been demonstrated along the thesis, at least in the particular cases of forest fire detection and monitoring. The application of the previous multi-robot perception techniques for fire detection and localization and the system for cooperatively fire monitoring presented in Chapter 6 are believed to be novel.

Several additional issues should be discussed. One of them is the scalability of the proposed approaches. The experiments shown in Chapters 6 and 8 can be considered close to the operational conditions, although on a lower scale. In order to apply the techniques in real situations, UAVs with higher endurance are required, mainly for the task of fire detection over wide areas. Nowadays there are UAVs, mainly developed in the military field, that have the required endurance, and therefore the extension of the techniques for their use by environment management agencies seems affordable. For instance, in the FiRE project (Ambrosia, 2002) a Predator UAV is used as a platform for forest fire surveillance.

Another issue is the particular UAV platform to be used in different applications. The experiments have shown mainly helicopters for fire detection tasks. However, it seems clear that, for this kind of applications, airplanes are more suitable. Also airships can be considered, although they should be big enough in order to withstand winds. Recently, stratospheric airships are gaining importance for this kind of applications. On the other hand, UAVs with hovering capabilities like helicopters are quite suitable for tasks like monitoring and inspection, as they can be positioned at will to obtain detailed views of the objects of interest. Nevertheless, the techniques are not dependant on the type of UAVs employed.

Finally, the thesis presents the use of fleets of UAVs in these tasks. One potential question is if the use of several UAVs offers advantages against the use of one complex UAV. For the task of fire detection (or in general, for exploration activities), a fleet of robots adds two different advantages. Firstly, the time needed to cover wide areas can be reduced, even with a superlinear gain with the number of UAVs. Also, as the thesis has shown, the cooperation of UAVs can be used to reduce the false alarm ratio, which is one of the main problems with currently employed systems (Arrue et al., 2000).

For the case of fire monitoring, employing several UAVs allows obtaining different and complementary views. In this case, even small UAVs could be employed by fire brigades for, at least, obtain views of areas difficult to be accessed. Also, several cheap UAVs can carry a bench of sensors that, on the other hand, would require an usually more expensive UAV with higher payload.

Nevertheless, along the work developed in the thesis and in discussions with different reviewers of the projects, the question came up of whether *autonomy* is required at all for this kind of tasks. The answer is that it depends. Teleoperated vehicles can be very helpful for certain tasks, like inspection of buildings, and in fact some of the results presented are obtained with a teleoperated helicopter. In any case, autonomous perception functions are required if no links or not enough bandwidth is available. Also, out-of-sight flights require autonomous navigation capabilities.

If the UAVs have to cover wide areas, then at least operational autonomy is required. The

importance of higher degrees of autonomy is more evident when considering fleets with a high number of robots. Controlling the fleet would require quite complex control centres or quite an amount of people if the robots are not endowed with decisional capabilities. In this case, also decentralized autonomous perception functions are required in order to reduce the computational power needed to process the huge amount of data from the different robots of the team.

Vision-based UAV navigation

Chapter 7 has shown a visual odometer for UAV motion estimation. There are many successful projects considering VTOL vehicles with autonomous navigation capabilities. Most of them work in outdoor environments and are based on GPS receivers and IMUs for pose estimation and control. These approaches work quite well, and therefore one important question is the necessity of developing navigation systems based on other sensors, in this case vision ones.

UAVs can also be very helpful in urban settings, for instance for applications like cable monitoring (Del Cerro et al., 2002), building inspection, surveillance and others. However, the use of UAV in those settings requires to increase the autonomy of UAVs in order to navigate when no GPS signal is available. Other applications require to position the vehicle with respect to objects of interest. For instance, for visual inspections of buildings, cinematography, etc. Indeed, helicopter pilots usually perceive as something very helpful to be able to lock a vision payload onto a target so that both, helicopter and payload, can be jointly controlled to maintain the target in the field of view.

The visual odometer presented can be employed for visual navigation under certain circumstances. It is based on feature-matching and results has been obtained for UAV pose estimation for landing, navigation and relative position estimation with respect to a building. Landing is usually a critical issue that requires some exteroceptive sensor, like ultrasound or laser rangefinders. A system based on vision can be more flexible in some approaches. Some authors have presented vision-based systems for landing, as in (Garcia-Pardo et al., 2001; Shakernia et al., 2002). They usually rely on artificial patterns. The techniques presented in Chapter 7 rely on natural landmarks and do not require artificial landing marks, although they could be considered as well. Moreover, the visual odometer can provide an estimation of the error committed, and then, it could be included in a more complex localization algorithm. Vision could be very helpful for landing on mobile platforms, like ships.

Moreover, in a more recent application in collaboration with researchers of the University of Linköping (Sweden), the same ideas can be used to obtain estimations of the relative position of other companions if features can be matched from different views. These relative poses are required for cooperative perception activities in order to anchor the objects to a common frame in which perform effectively information fusion.

9.1.3 The demonstrations

Besides simulations, many of the ideas presented in this thesis have been actually tested online during the field experiments described in Chapter 8, and others have been applied on real data obtained in these experiments.

In the frame of the COMETS project, part of the ideas presented in Chapters 3 and 4 were tested on line during the final experiments described in Chapter 8, in May 2005. Also, data from these experiments have been used to test the algorithms presented in Chapter 5.

Moreover, the algorithms of Chapter 6 have been tested online in several occasions in the experiments carried out in the SPREAD European project. The extension to UAVs was tested using recorded data from conventional helicopters (see Chapter 6), and also online in the general experiments of the COMETS project, with real UAVs, although in this case the fire monitoring experiments were performed with very small fires in which no significant evolution of the fire front was measurable. Nevertheless, the experiments have shown the feasibility for online fire monitoring employing small sized VTOLs.

The techniques of Chapter 7 have been applied to data gathered in those experiments, and also in other local experiments carried out in the facilities of the Escuela Superior de Ingenieros, for building inspection and thermal leakage detection.

9.2 Perspectives and future work

The following paragraphs describe some work currently ongoing plus some personal views about the perspectives of the techniques presented in the thesis.

9.2.1 Active cooperative perception

A complete cooperative perception system not only considers information fusion, but also the development of actions. As commented in Chapter 8, in the COMETS experiments the loop between perception and action for the whole UAV fleet was closed at the planning level in a central station, as the UAVs only had operational autonomy. This central station, from the results provided by the multi-UAV perception system, would perform the adequate actions when new events were detected, like sending new tasks to the UAVs, etc.

However, the perception and decisional architectures developed in COMETS allow considering more decentralized schemes for cooperation. The higher levels of the decisional architecture presented by Gancet (2005) and Gancet et al. (2005) can be fused with the decentralized perception functions, allowing for coordination and cooperation for information gathering activities. In the AEROSENS project, an extension of a similar decisional architecture, tightly coupled with the perception activities is being developed. It will allow to coordinate information gathering tasks, taking into account the expected information gain. Market-based approaches for decentralized task allocation will be considered, employing the belief states and measurement models to determine the utility and bids associated to the tasks.

Moreover, in the frame of the URUS project, planning under uncertainty by using POMDPs approaches will be considered (Section 2.6). POMDPs allows to optimally consider perception and action within the same Bayesian framework. An interesting implication that is worth to be analyzed is given by the fact that if the decisions of the individual robots of the fleet are based on their local belief states (for instance, employing a MDP or POMDP approach), the sharing of information

Figure 9.1: Left: Crossbow Node. Middle: a node with a dragonfly on the antenna for scale comparison. Right: helicopter carrying a node.

will lead indirectly to some coordination on the fleet. Very preliminary results have been already commented in Chapter 4. Some researchers have shown these effects (Grocholsky et al., 2003), but additional research will be conducted.

9.2.2 Complex environment representations

Another important issue considered in this thesis is the fusion of information of different nature, at different levels of abstraction. Usually, an object is not only described by its position, but other information as well. For instance, for the application of forest fire detection not only the position of the fire is important, but also to be sure that the considered object is actually a fire or belongs to another class of object.

In non-structured environments, tracking the evolution of an object requires considering different features in order to solve the data association ambiguities, especially in the presence of clutter. Therefore, probabilistic vision algorithms should consider likelihood models for different aspects in the feature extraction process, as identity, appearance belief states and others

Here, only the issues of classification and identity have been covered. In the frame of the URUS project, cooperative tracking of persons in urban environments is considered. Complex representations considering appearance information, shape and others will have to be computed locally and shared among the robots of the fleet for solving this task.

9.2.3 Networked robots

The field of *Network Robots Systems* (NRS) deals with teams of robots that can communicate among themselves, with sensors embedded in the environment (like Wireless Sensor Networks -WSNs-, see Fig. 9.1) and with humans.

The use of a NRS for service robotics is considered in the URUS project. In this project, a fleet of networked robots will offer services like guiding, transporting of people and/or goods, surveillance in an urban scenario. These services require functionalities like human tracking, event detection, etc. in very dynamic environment. The extension of the developed techniques for cooperative and

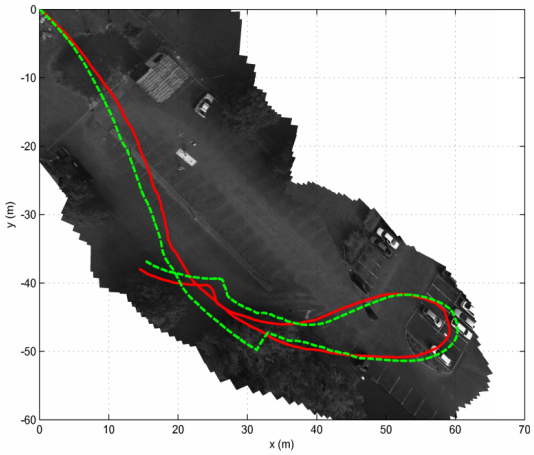

Figure 9.2: Vision-based estimated motion corrected with a mosaic (mosaic and results generated by Fernando Caballero).

decentralized human assistance, like guidance, tracking, surveillance, will be considered.

As commented above, these complex scenarios will require more complex environment models, in which information at different levels is shared and employed for data association. Moreover, it is very important to incorporate information provided by humans for service robotics.

9.2.4 UAV vision-based navigation

The limitations of the algorithms proposed for visual navigation with UAVs were pointed out above. Nevertheless, these algorithms provide estimations on the relative motion of one UAV. With the decreasing cost of UAV airframes and technologies it is quite likely that, in the short future, small UAVs will be used even for indoor applications, for which localization based on other means than GPS is required.

On the other hand, vision-based localization capabilities for UAVs could be needed in some applications. The work presented in Chapter 7 has been carried out closely with Fernando Caballero. This work is being extended, adding to the motion estimation algorithms presented (basically a kind of odometer) mechanisms in order to correct the drift associated (Caballero et al., 2006; Caballero et al., 2007) (Fig. 9.2). The final goal is to add a second level of processing in charge of localization and mapping of landmarks, in a kind of SLAM approach.

Appendix A

Multivariate Gaussian Distribution: properties

A.1 Multivariate Gaussian Distribution

The multivariate Gaussian distribution is parameterized by its mean $\boldsymbol{\mu}$ and the covariance matrix $\boldsymbol{\Sigma}$. The pdf of a multivariate Gaussian is given by:

$$p(\mathbf{X}) = \frac{1}{(\sqrt{2\pi})^N \sqrt{|\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{X}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\mathbf{X}-\boldsymbol{\mu})} \tag{A.1}$$

The mean $\boldsymbol{\mu}$ is the value of $\mathbf{X}$ where $p(\mathbf{X})$ reaches its maximum value. The previous distribution is also denoted as $p(\mathbf{X}) = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.

An alternative parametrization of the Gaussian distribution is the canonical form, that employs as parameters the information vector $\boldsymbol{\xi} = \boldsymbol{\Sigma}^{-1}\mu$ and information matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$.

$$p(\mathbf{X}) = \frac{\sqrt{|\boldsymbol{\Omega}|}}{(\sqrt{2\pi})^N} e^{-\frac{1}{2}\mathbf{X}^T\boldsymbol{\Omega}\mathbf{X}+\mathbf{X}^T\boldsymbol{\xi}} \tag{A.2}$$

If a second random variable $\mathbf{Y}$ is a linear transformation of $\mathbf{X}$, $\mathbf{Y} = \mathbf{A}\mathbf{X} + \mathbf{b}$, then $p(\mathbf{Y}) = \mathcal{N}(\mathbf{A}\boldsymbol{\mu} + \mathbf{b}, \mathbf{A}\boldsymbol{\Sigma}\mathbf{A}^T)$

A.2 Marginalization and conditioning of a Multivariate Gaussian

The covariance matrix form and canonical form for the multivariate Gaussian distribution present orthogonal algebraic characteristics with respect to the conditioning and marginalization operations for multivariate distributions.

$$\boldsymbol{\Omega} = \begin{pmatrix} \boldsymbol{\Omega}_{aa} & \boldsymbol{\Omega}_{ab} \\ \boldsymbol{\Omega}_{ba} & \boldsymbol{\Omega}_{bb} \end{pmatrix} \quad \boldsymbol{\xi} = \begin{pmatrix} \boldsymbol{\xi}_{aa} \\ \boldsymbol{\xi}_{bb} \end{pmatrix} \tag{A.3}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{aa} & \boldsymbol{\Sigma}_{ab} \\ \boldsymbol{\Sigma}_{ba} & \boldsymbol{\Sigma}_{bb} \end{pmatrix} \boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_{aa} \\ \boldsymbol{\mu}_{bb} \end{pmatrix} \tag{A.4}$$

	Marginalization $p(\mathbf{A}) = \int p(\mathbf{A}, \mathbf{B}) d\mathbf{B}$	Conditioning $p(\mathbf{A}\|\mathbf{b}) = p(\mathbf{A}, \mathbf{b})/p(\mathbf{b})$
Information Form	$\bar{\boldsymbol{\Omega}}_{aa} = \boldsymbol{\Omega}_{aa} - \boldsymbol{\Omega}_{ab}\boldsymbol{\Omega}_{bb}^{-1}\boldsymbol{\Omega}_{ba}$ $\bar{\boldsymbol{\xi}}_{aa} = \boldsymbol{\xi}_{aa} - \boldsymbol{\Omega}_{ab}\boldsymbol{\Omega}_{bb}^{-1}\boldsymbol{\xi}_{bb}$	$\bar{\boldsymbol{\Omega}}_{a\|b} = \boldsymbol{\Omega}_{aa}$ $\bar{\boldsymbol{\xi}}_{a\|b} = \boldsymbol{\xi}_{aa} - \boldsymbol{\Omega}_{ab}\mathbf{b}$
Covariance Form	$\bar{\boldsymbol{\Sigma}}_{aa} = \boldsymbol{\Sigma}_{aa}$ $\bar{\boldsymbol{\mu}}_{aa} = \boldsymbol{\mu}_{aa}$	$\bar{\boldsymbol{\Sigma}}_{a\|b} = \boldsymbol{\Sigma}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}\boldsymbol{\Sigma}_{ba}$ $\bar{\boldsymbol{\mu}}_{a\|b} = \boldsymbol{\mu}_{aa} - \boldsymbol{\Sigma}_{ab}\boldsymbol{\Sigma}_{bb}^{-1}(\mathbf{b} - \boldsymbol{\mu}_{bb})$

A.3 Kullback-Leibler Divergence

The Kullback-Leibler divergence (also called relative entropy)(Kullback and Leibler, 1951) is used in information theory as a measure of the difference between a "true" or reference distribution $p(\mathbf{X})$ and a proposed distribution $q(\mathbf{X})$. It is usually used to measure the "distance" between distributions, although it is not a distance. The Kullback-Leibler divergence is defined as:

$$D_{KL}(p\|q) = \int p(\mathbf{X}) log \frac{p(\mathbf{X})}{q(\mathbf{X})} d\mathbf{X} \tag{A.5}$$

for the continuous case.

The KL divergence of two N-dimensional Gaussians with covariance matrices $\boldsymbol{\Sigma}_p$ and $\boldsymbol{\Sigma}_q$ is:

$$2D_{KL}(p\|q) = tr(\boldsymbol{\Sigma}_q^{-1}\boldsymbol{\Sigma}_p) + log|\boldsymbol{\Sigma}_q| - log|\boldsymbol{\Sigma}_p| + (\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^T\boldsymbol{\Sigma}_q^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) - N \tag{A.6}$$

Appendix B

Infrared and visual image fusion for fire detection

The main application presented in this thesis is forest fires detection and monitoring. This appendix presents fire segmentation algorithms using infrared and visual images. Moreover, it presents a method to combine, under certain circumstances, an infrared and a visual camera to obtain more robust results on fire detection.

B.1 Fire segmentation in colour images

The objective of fire segmentation is to separate fire objects from the background on the image plane. The approach presented here is based on a supervised learning phase in which a probability histogram over the RGB colour space is built from image patches with fire. Afterwards, this histogram is employed in order to classify an object as fire or not employing colour as the main feature.

The training phase consists of the following steps:

1. Creation of pairs of training images: each pair consists of a colour image I and a Boolean mask B that specifies the locations at which the target object occurs. It is necessary to ensure a wide range of training pairs to obtain a generic fire colour histogram.

2. Building of the colour histogram: for every image I_i and for every pixel location in the image, if the value in the corresponding mask location is "true" then a Gaussian distribution is added to the colour histogram centred at the colour value that corresponds to the colour of the individual pixel. Otherwise, a smaller Gaussian distribution is subtracted from the colour histogram.

Algorithm B.1 summarizes the procedure. Once the histogram is built, a binary histogram is obtained by using a threshold over the previously constructed smoothed histogram. This binary histogram is a function which returns if a given colour belongs to the fire region. The theshold is selected using an iterative process, so that the threshold leads to the minimum classification error.

Algorithm B.1 histogram ← Training fire $(\{I_i, B_i\}, i = 1, \ldots, N)$

```
h = 0
for all Image training pair {I_i, B_i} do
  for all Pixel m = [u, v]^T in I_i do
    c = [r(m), g(m), b(m)]^T
    if B(m) = 1 then
      Add N(c, Σ_{c,true}) to h
    else
      Substract N(c, Σ_{c,false}) to h
    end if
  end for
end for
```

B.2 Fire segmentation in infrared images

Another important sensor employed by the UAVs of the fleet considered in this thesis are infrared cameras. Infrared cameras can be classified in two types: thermal cameras and non-thermal cameras. Both types of cameras measure the radiation intensity throughout the scene. Additionally, thermal cameras include software tools that transform radiation intensity into temperature estimations by using the emissivity indices of the materials and other physical parameters. For thermal cameras, simple temperature thresholds are used to distinguish fire and background pixels.

In the images provided by non-thermal infrared cameras, each pixel is qualitatively represented by a grey level, where brighter levels correspond to higher radiation. For these images, the segmentation approach is to use a thresholding technique. The technique adopted for these images has been developed by Martínez de Dios, and is described in (Martínez de Dios and Ollero, 2004). The main idea of the method is to extract the particularities of a computer vision application during a training phase and use them to supervise a coarse-to-fine multiresolution histogram analysis. The technique is based on the commonly accepted assumption that considers that the intensity values of the pixels belonging to the same object are distributed in histogram modes or clusters.

In the training stage, a set of training images and their corresponding desired threshold values are used to identify the conditions under which a histogram mode is considered to be originated by the object. These particularities are used to train a fuzzy supervision system via the ANFIS algorithm (Jang, 1993). In the application stage, the trained system is used to supervise the classification of the histogram modes in those originated by the object or by the background. Finally, the threshold value is computed as the value that optimally distinguishes between the object-classified and background-classified modes. A more detailed description can be found in (Martínez de Dios and Ollero, 2004).

B.3 Infrared and visual image fusion for fire detection

In the application of fire detection, infrared images are the preferred sensor due to their detection capabilities. However, they usually present relatively high false alarm ratios, as other heated objects could produce similar signatures in infrared images. On the other hand, the algorithm for visual

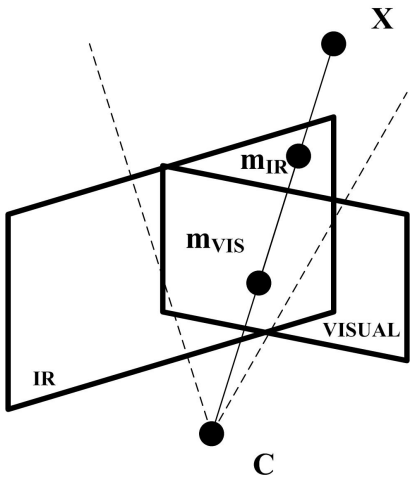

Figure B.1: Epipolar geometry of camera configuration.

images presents a lower false alarm ratio, but it has worse detection capabilities.

There is an interesting configuration in which two cameras of different modalities can be combined to obtain more robust results on fire segmentation. The Heliv helicopter that participated in the COMETS experiments, described in Chapter 8, carries on board two cameras, one visual video camera and a low-cost OEM non-thermal video camera. Figure 8.9 shows the configuration.

This section presents how the images from both cameras can be combined to obtain a kind of multi-spectral camera.

B.3.1 Registration of infrared and visual images

If both cameras of the system of Fig. 8.9 share the centre of projection, then the geometry of this configuration is described by Fig. B.1. $\mathbf{m}_{IR} = \begin{pmatrix} u & v & 1 \end{pmatrix}^T$ and $\mathbf{m}_{vis} = \begin{pmatrix} u' & v' & 1 \end{pmatrix}^T$ are the images, at the same instant, of a point $\mathbf{X}$ in homogeneous pixel coordinates on the infrared and visual camera images respectively.

If the centres of projection of both cameras are assumed to be coincident (point $\mathbf{C}$), the relation between both images is given by:

$$s\mathbf{m}_{IR} = \mathbf{H}_\infty \mathbf{m}_{vis} \tag{B.1}$$

where $\mathbf{H}_\infty$ is a 3×3 matrix called the infinity homography and s is a scale factor. As shown in Section 7.2.1:

$$\mathbf{H}_\infty \simeq \mathbf{A}_{IR}\mathbf{R}\mathbf{A}_{vis}^{-1} \tag{B.2}$$

where $\mathbf{A}_{IR}$ and $\mathbf{A}_{vis}$ are the internal calibration matrices of both cameras, and $\mathbf{R}$ is the rotation matrix that relates the camera-centered coordinate frames. Thus, $\mathbf{H}_\infty$ could be computed if the cameras are calibrated and the relative frame transformation is known. However, $\mathbf{H}_\infty$ can be also

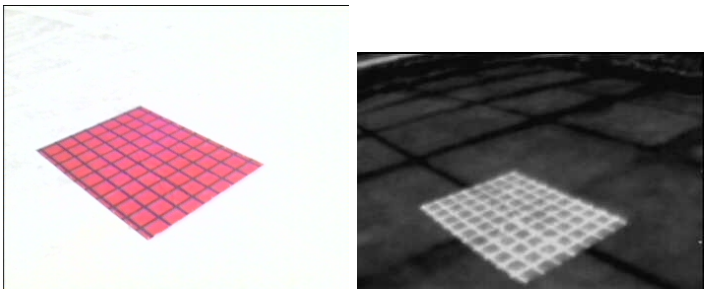

Figure B.2: Calibration pattern for relating infrared and visual images.

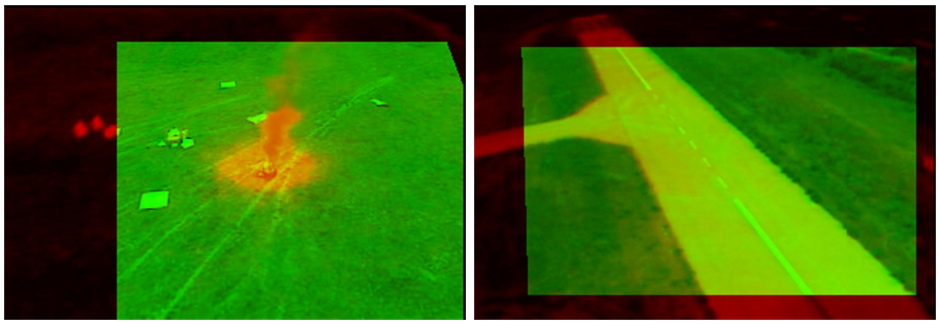

Figure B.3: Two examples of the resultant combined images. The image is obtained filling the red field with the infrared pixel intensities and the green field with the visual image.

calculated knowing at least four correspondences among points or lines in both images by using the algorithms presented in Appendix C.

There are many algorithms for point matching between images of the same modality. However, this is a challenging problem when dealing with images of different modalities. For the experiments presented in this thesis, the calibration has been done using a known pattern that is visible on both types of cameras, as shown in Fig. B.2. It should be noticed that the relation $\mathbf{H}_\infty$ has to be computed only once, provided that the relative orientation of the cameras and their internal calibration do not change. Caspi And Irani (2002) have presented another method that can be used to compute $\mathbf{H}_\infty$ from measurements on the image plane, without the need of a calibration grid.

In the system considered, the centres of projection of both cameras will not be actually coincident, but the equations above hold if the distance between the centres of projection is small compared with the distances of the points of the scene with respect to the cameras. Figure B.3 shows some results of the combination of infrared and visual images.

Once $\mathbf{H}_\infty$ is known, both images can be registered, and thus, the detected alarms on the visual image can be transformed to the infrared image plane.

Table B.1: Characterization of algorithms

Algorithm	P_D	P_F
Infrared	0.962	0.045
Visual	0.819	0.023
Combined	0.981	0.003

B.4 Combined fire detection

The operation of the fire detection system consists in applying the algorithms presented above to infrared and visual images. Each algorithm will reach at a decision z for each pixel stating that there is fire ($z = 1$) or not ($z = 0$) in that position. By using $\mathbf{H}_\infty$, both measurements can be expressed in a common frame (see Fig. B.3). Thus, for each pixel, there is a set of measurements $\mathbf{z} = \begin{pmatrix} z_{IR} & z_{VIS} \end{pmatrix}^T$. The objective is to obtain a fusion rule that leads to a fused decision d for the global hypothesis T_0 (not fire) or T_1 (fire) for each pixel. A maximum likelihood approach can be used to select the final hypothesis. In (Gunatilaka and Baertlein, 2001) it can be seen that the likelihood ratio lr given by:

$$lr = \frac{p(\mathbf{z}|T_1)}{p(\mathbf{z}|T_0)} \tag{B.3}$$

can be used as a fusion rule that leads to the minimum Bayes risk. The objective is to find an optimum threshold th so that if $lr > th$, then $d = 1$ (that is, hypothesis T_1, fire, is selected). Otherwise, $d = 0$. Assuming that the measurements obtained from both cameras are conditionally independent, (B.3) leads to:

$$lr = \frac{p(z_{IR}|T_1)p(z_{VIS}|T_1)}{p(z_{IR}|T_0)p(z_{VIS}|T_0)} \tag{B.4}$$

Each of the sensors can be characterized by their probabilities of correct detection $P_{D,i} = p(z_i = 1|T_1)$ and false positive $P_{F,i} = p(z_i = 1|T_0)$. The likelihood ratio is usually expressed in terms of the log-likelihood. Taking logarithms,

$$\log lr = \sum_{i \in \{IR, vis\}} \left[z_i log(\frac{P_{D,i}}{P_{F,i}}) + (1 - z_i) log(\frac{1 - P_{D,i}}{1 - P_{F,i}}) \right] \tag{B.5}$$

Table B.1 shows the classification results obtained by the algorithms alone and the combined algorithm.

Appendix C

Homographies: computation and error estimation

C.1 Computation of homographies from matches

A homography (Hartley and Zisserman, 2004) is any linear and invertible mapping from the projective space $\mathbb{P}^2$ into $\mathbb{P}^2$. It is represented by a 3×3 invertible matrix defined up to scale factor, so it is composed by eight independent parameters.

If two image points are related by a homography $\mathbf{H}$, then

$$s_i \mathbf{m}'_i = \mathbf{H}\mathbf{m}_i \tag{C.1}$$

where $\mathbf{m}_i = \begin{pmatrix} u_i & v_i & 1 \end{pmatrix}^t$ is the image point in homogenous coordinates, $\mathbf{H}$ is a homography matrix and s is a scale factor.

$$\mathbf{H} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} = \begin{pmatrix} \mathbf{h}_1^T \\ \mathbf{h}_2^T \\ \mathbf{h}_3^T \end{pmatrix} \tag{C.2}$$

where $\mathbf{h}_i^T$ is the i-th row of $\mathbf{H}$.

Given a set of matches $\{\mathbf{m}_i \leftrightarrow \mathbf{m}'_i\}$, the objective is to obtain the homography $\mathbf{H}$ that relates them. In order to do that, it should be noticed that the vectors $\mathbf{m}'_i$ and $\mathbf{H}\mathbf{m}_i$ are equal up to a scale factor s_i (they are expressed in homogeneous coordinates). Therefore, the cross-product $\mathbf{m}'_i \times \mathbf{H}\mathbf{m}_i$ equals zero.

$$\mathbf{m}'_i \times \mathbf{H}\mathbf{m}_i = \mathbf{m}'_i \times \begin{pmatrix} \mathbf{h}_1^T \mathbf{m}_i \\ \mathbf{h}_2^T \mathbf{m}_i \\ \mathbf{h}_3^T \mathbf{m}_i \end{pmatrix} = \begin{pmatrix} v'_i \mathbf{h}_3^T \mathbf{m}_i - \mathbf{h}_2^T \mathbf{m}_i \\ \mathbf{h}_1^T \mathbf{m}_i - u'_i \mathbf{h}_3^T \mathbf{m}_i \\ u'_i \mathbf{h}_2^T \mathbf{m}_i - v'_i \mathbf{h}_1^T \mathbf{m}_i \end{pmatrix} = \mathbf{0} \tag{C.3}$$

Reordering terms, the previous equation can be written as:

$$\begin{pmatrix} \mathbf{0}^T & -\mathbf{m}_i^T & v_i'\mathbf{m}_i^T \\ \mathbf{m}_i^T & \mathbf{0}^T & -u_i'\mathbf{m}_i^T \\ -v_i'\mathbf{m}_i^T & u_i'\mathbf{m}_i^T & \mathbf{0}^T \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = \mathbf{0} \tag{C.4}$$

This is a linear set of equations on the elements of $\mathbf{H}$, stored in the 9-vector $\mathbf{h} = \begin{pmatrix} \mathbf{h}_1^T & \mathbf{h}_2^T & \mathbf{h}_3^T \end{pmatrix}^T$. Each pair of point correspondences contributes with two equations, as the third row in (C.4) is a linear combination of the other two.

$$\begin{pmatrix} \mathbf{0}^T & -\mathbf{m}_i^T & v_i'\mathbf{m}_i^T \\ \mathbf{m}_i^T & \mathbf{0}^T & -u_i'\mathbf{m}_i^T \\ & \dots & \end{pmatrix} \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix} = \mathbf{0} \Rightarrow \mathbf{A}\mathbf{h} = \mathbf{0} \tag{C.5}$$

Therefore, to compute $\mathbf{H}$ from the set of correspondences, the linear system $\mathbf{A}\mathbf{h} = \mathbf{0}$ must be solved. As the homography has eight degrees of freedom (is defined up to a scale factor), knowing only four correspondences allows to determine $\mathbf{H}$. In general, the process of obtaining a homography considers matches between a set of points $\mathbf{M} = \{\mathbf{m}_i, i = 1, \dots, N\}$ and $\mathbf{M'} = \{\mathbf{m}_i', i = 1, \dots, N\}$, with $N > 4$. The solution is then to obtain the vector $\mathbf{h}$ that minimizes $\|\mathbf{A}\mathbf{h}\|$ subject to $\|\mathbf{h}\| = 1$. The solution to this (Golub and Loan, 1996; Hartley and Zisserman, 2004) is the unit eigenvector of $\mathbf{A}^T\mathbf{A}$ corresponding to the smallest eigenvalue.

C.2 Estimation of homography covariances

Using the procedure described, an estimation $\hat{\mathbf{h}}$ of the homography can be obtained. The matches between points are usually affected by errors. The errors of the matches $\mathbf{m}_i'$ are represented by zero-mean Gaussian distribution with covariance matrix $\mathbf{\Sigma}_{\mathbf{M}'}$.

It can be seen in (Hartley and Zisserman, 2004) that the covariance matrix $\mathbf{\Sigma}_{\mathbf{h}}$ associated to the estimated parameters $\hat{\mathbf{h}}$ using the procedure described above is given by:

$$\mathbf{\Sigma}_{\mathbf{h}} = (\mathbf{J}^T \mathbf{\Sigma}_{\mathbf{M}'}^{-1} \mathbf{J})^{\dagger} \tag{C.6}$$

where $\mathbf{J}$ is the Jacobian matrix $\partial \mathbf{m}_i'/\partial \mathbf{h}$ evaluated at $\hat{\mathbf{h}}$ and $\dagger$ is the pseudo-inverse [1]. Each match $\{\mathbf{m}_i' \leftrightarrow \mathbf{m}_i\}$ contributes to the matrix $\mathbf{J}$ with a block:

$$\mathbf{J}_i = \frac{1}{s_i'} \begin{pmatrix} \mathbf{m}_i{}^T & \mathbf{0}^T & -u_i'\mathbf{m}_i^T \\ \mathbf{0}^T & \mathbf{m}_i^T & -v_i'\mathbf{m}_i^T \end{pmatrix} \tag{C.7}$$

where $s_i' = \mathbf{h}_3^T \mathbf{m}_i$. The complete Jacobian is $\mathbf{J} = (\mathbf{J}_1{}^T, \dots, \mathbf{J}_i{}^T, \dots, \mathbf{J}_N{}^T)^T$.

If the errors committed in the matches are independent, then the covariance matrix of the errors in the matches is block-diagonal $\mathbf{\Sigma}_{\mathbf{M}'} = diag(\mathbf{\Sigma}_{\mathbf{m}_i'})$, and (C.6) becomes:

[1] The pseudo-inverse of a $m \times n$ matrix $\mathbf{A}$ of rank n is $\mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$

$$\mathbf{\Sigma_h} = (\sum_i \mathbf{J}_i{}^T \mathbf{\Sigma}^{-1}_{\mathbf{m}'_i} \mathbf{J}_i)^\dagger \tag{C.8}$$

C.3 Transformation of homography covariances

Finally, it is interesting to have expressions for the transformation of the 9×9 covariance matrix $\mathbf{\Sigma_h}$ on the computed homography, when a linear transformation is applied over it.

C.3.1 Linear operations on homographies

As seen in Appendix A, given a vector $\mathbf{p}$ with covariance matrix $\mathbf{\Sigma_p}$, then the vector $\mathbf{p}_{new} = \mathbf{Ap}$ has covariance matrix $\mathbf{\Sigma}_{\mathbf{p}_{new}} = \mathbf{A\Sigma_p A}^T$.

In general, the operations are performed with the homography in matrix form. Given a 3×3 matrix $\mathbf{B}$, with elements b_{ij}, the relation $\mathbf{H}_{new} = \mathbf{HB}$ becomes (in vector form):

$$\mathbf{h}_{new} = \underbrace{\begin{pmatrix} \mathbf{B}^T & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{B}^T \end{pmatrix}}_{\mathbf{B}_v} \mathbf{h} \tag{C.9}$$

On the other hand, if the relation is like $\mathbf{H}_{new} = \mathbf{BH}$, then

$$\mathbf{h}_{new} = \begin{pmatrix} b_{11} & 0 & 0 & b_{12} & 0 & 0 & b_{13} & 0 & 0 \\ 0 & b_{11} & 0 & 0 & b_{12} & 0 & 0 & b_{13} & 0 \\ 0 & 0 & b_{11} & 0 & 0 & b_{12} & 0 & 0 & b_{13} \\ b_{21} & 0 & 0 & b_{22} & 0 & 0 & b_{23} & 0 & 0 \\ 0 & b_{21} & 0 & 0 & b_{22} & 0 & 0 & b_{23} & 0 \\ 0 & 0 & b_{21} & 0 & 0 & b_{22} & 0 & 0 & b_{23} \\ b_{31} & 0 & 0 & b_{32} & 0 & 0 & b_{33} & 0 & 0 \\ 0 & b_{31} & 0 & 0 & b_{32} & 0 & 0 & b_{33} & 0 \\ 0 & 0 & b_{31} & 0 & 0 & b_{32} & 0 & 0 & b_{33} \end{pmatrix} \mathbf{h} \tag{C.10}$$

equivalently:

$$\mathbf{h}_{new} = \underbrace{\begin{pmatrix} b_{11}\mathbf{I} & b_{12}\mathbf{I} & b_{13}\mathbf{I} \\ b_{21}\mathbf{I} & b_{22}\mathbf{I} & b_{23}\mathbf{I} \\ b_{31}\mathbf{I} & b_{32}\mathbf{I} & b_{33}\mathbf{I} \end{pmatrix}}_{\mathbf{B}_v} \mathbf{h} \tag{C.11}$$

The helpful relation $\mathbf{H}_{new} = \mathbf{B}^{-1}\mathbf{HB}$ becomes in vector form:

$$\mathbf{h}_{new} = \underbrace{\begin{pmatrix} b'_{11}\mathbf{B}^T & b'_{12}\mathbf{B}^T & b'_{13}\mathbf{B}^T \\ b'_{21}\mathbf{B}^T & b'_{22}\mathbf{B}^T & b'_{23}\mathbf{B}^T \\ b'_{31}\mathbf{B}^T & b'_{32}\mathbf{B}^T & b'_{33}\mathbf{B}^T \end{pmatrix}}_{\mathbf{B}_v} \mathbf{h} \tag{C.12}$$

where b'_{ij} are the elements of $\mathbf{B}^{-1}$.

These matrixes can be employed to estimate the transformed covariance matrix $\mathbf{\Sigma}_{\mathbf{h}_{new}} = \mathbf{B}_v \mathbf{\Sigma}_{\mathbf{h}} \mathbf{B}_v^T$.

C.3.2 Composition of homographies

In some applications, like those presented in Chapter 7, it is required to chain homographies that relate different images. The expressions are of the form:

$$\mathbf{H}_{0i} = \mathbf{H}_{(i-1)i}\mathbf{H}_{0(i-1)} \tag{C.13}$$

It can be seen under some manipulation that the Jacobians of this equation, expressed in vector form, are of the form:

$$\frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{(i-1)i}} = \begin{pmatrix} \mathbf{H}^T_{0(i-1)} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}^T_{0(i-1)} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{H}^T_{0(i-1)} \end{pmatrix} \tag{C.14}$$

$$\frac{\partial \mathbf{h}_{0i}}{\partial \mathbf{h}_{0(i-1)}} = \begin{pmatrix} h_{11}\mathbf{I} & h_{12}\mathbf{I} & h_{13}\mathbf{I} \\ h_{21}\mathbf{I} & h_{22}\mathbf{I} & h_{23}\mathbf{I} \\ h_{31}\mathbf{I} & h_{32}\mathbf{I} & h_{33}\mathbf{I} \end{pmatrix} \tag{C.15}$$

Bibliography

Alami, R., Chatila, R., Fleury, S., Ghallab, M., and Ingrand, F. (1998a). An Architecture for Autonomy. *The International Journal of Robotics Research*, 17(4):315–337.

Alami, R., Fleury, S., Herrb, M., Ingrand, F., and Robert, F. (1998b). Multi-robot cooperation in the MARTHA project. *IEEE Robotics and Automation Magazine*, 5(1):36–47.

Amano, H., Osuka, K., and Tzyh-Jong, T. (2001). Development of vertically moving robot with gripping handrails for fire fighting. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 2, pages 661–667.

Ambrosia, V. (2002). Remotely piloted vehicles as fire imaging platforms: The future is here! *Wildfire Magazine*.

Amidi, O. (1996). *An autonomous vision-guided helicopter*. PhD thesis, Carnegie-Mellon University.

Amidi, O., Kanade, T., and Fujita, K. (1999). A visual odometer for autonomous helicopter flight. *Journal of Robotics and Autonomous Systems*, 28:185 – 193.

Arrue, B., Ollero, A., and Martínez de Dios, J. (2000). An intelligent system for false alarm reduction in infrared forest-fire detection. *IEEE Intelligent Systems*, 15(3):64–73.

Asama, H., Matsumoto, A., and Ishida, Y. (1989). Design of an autonomous and distributed robot system: ACTRESS. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 283–290.

Asif, A. and Moura, J. (2005). Block matrices with L-block-banded inverse: inversion algorithms. *IEEE Transactions on Signal Processing*, 53(2):630–642.

Ayache, N. and Sander, P. T. (1991). *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*. The MIT Press, Cambridge, MA, USA.

Balch, T. and Arkin, R. (1998). Behavior-based formation control for multirobot teams. *IEEE Transactions on Robotics and Automation*, 14(6):926–939.

Balch, T. and Parker, L. (2002). *Robot teams: from diversity to polymorphism*. A K Peters.

Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and Data Association*. Academic Press.

Bar-Shalom, Y. and Li, X. (1995). *Multitarget-multisensor tracking: principles and techniques*. YBS Publishing.

Bar-Shalom, Y., Li, X., and T.Kirubarajan (2001). *Estimation with Applications to Tracking and Navigation*. Wiley Interscience.

Bertuccelli, L. and How, J. (2006). Search for dynamic targets with uncertain probability maps. In *Proc. of the American Control Conference*, pages 737–742.

Bigün, J. and Granlund, G. H. (1987). Optimal orientation detection of linear symmetry. In *Proc. of the IEEE First International Conference on Computer Vision*, pages 433–438, London, Great Britain.

Blanco, J., Gonzalez, J., and Fernandez-Madrigal, J. (2007). A consensus-based approach for estimating the observation likelihood of accurate range sensors. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 4032–4037.

Bolić, M., Djurić, P., and Hong, S. (2005). Resampling algorithms and architectures for distributed particle filters. *IEEE Transactions on Signal Processing*, 53(7):2442–2450.

Bosch, S., Lacroix, S., and Caballero, F. (2006). Autonomous detection of safe landing areas for an UAV from monocular images. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5522–5527.

Bourgault, F. and Durrant-Whyte, H. (2004). Communication in general decentralized filters and the coordinated search strategy. In *Proc. of The 7th Int. Conf. on Information Fusion*, pages 723–730.

Bourgault, F., Furukawa, T., and Durrant-Whyte, H. (2004). Decentralized bayesian negotiation for cooperative search. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2681–2686.

Bradshaw, A. (1991). The UK security and fire fighting advanced robot project. In *IEE Colloquium on Advanced Robotic Initiatives in the UK*.

Bueno, S., Azinheira, J., Ramos, J., Paiva, E., Rives, P., Elfes, A., Carvalho, J., and Silveira, G. (2002). Project AURORA: Towards an autonomous robotic airship. In *Proc. of the Workshop WS6 Aerial Robotics*, pages 43–54. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS 2002.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21(3):376–386.

Caballero, F., Ferruz, J., and Ollero, A. (2004). Compensación del movimiento en secuencias de imágenes aéreas. In *XXV Jornadas de Automtica*.

Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2005). A visual odometer without 3D reconstruction for aerial vehicles. Applications to building inspection. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 4684–4689.

Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2006). Improving vision-based planar motion estimation for unmanned aerial vehicles through online mosaicing. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2860–2865. IEEE.

Caballero, F., Merino, L., Ferruz, J., and Ollero, A. (2007). Homography-based kalman filter for mosaic building. applications to UAV position estimation. In *Proc. of the IEEE International Conference on Robotics and Automation*. IEEE.

Cai, A., Fukuda, T., and Arai, F. (1997). Information sharing among multiple robots for cooperation in cellular robotic system. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 1768–1774.

Campbell, D., Born, W. G., Beck, J., Bereska, B., Frederick, K., and Hua, S. (2002). Airborne wildfire intelligence system: a decision support tool for wildland fire managers in Alberta. In *Proc. SPIE, Thermosense XXIV*, volume 4710, pages 159–170.

Cánovas, J., LeBlanc, K., and Saffiotti, A. (2004). Robust multi-robot object localization using fuzzy logic. In Springer-Verlag, editor, *Proc. of the International Robocup Symposium*.

Cao, Y., Fukunaga, A., and Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27.

Casbeer, D., Kingston, D., Bear, R., McLain, T., and Li, S. (2005). Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of System Science*, pages 1–18.

Caspi, Y. and Irani, M. (2002). Aligning non-overlapping sequences. *Int. J. Comput. Vision*, 48(1):39–51.

Chaimowicz, L., Cowley, A., Sabella, V., and Taylor, C. (2003). ROCI: a distributed framework for multi-robot perception and control. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 1, pages 266–271.

Chaimowicz, L., Grocholsky, B., Kellerand, J., Kumar, V., and Taylor, C. (2004). Experiments in multirobot air-ground coordination. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 4, pages 4053–4058.

Christophersen, H., Dhingra, M., Guily, R., Hart, M., Johnson, E., and Kahn, A. (2001). Development of an autonomous aerial reconnaissance system at Georgia Tech. *2001 Aerial Robotics Competition*.

Chuvieco, E. and Martin, P. (1994). A simple method for fire growth mapping using AVHRR channel 3 data. *International Journal of Remote Sensing*, 15:3141–3146.

Coates, M. (2004). Distributed particle filters for sensor networks. In *Proc. of the International symposium on Information Processing in Sensor Networks IPSN*, pages 99–107.

Collett, T., MacDonald, B., and Gerkey, B. (2005). Player 2.0: Toward a practical robot programming framework. In *Proc. of the Australasian Conference on Robotics and Automation.*, pages 317–323.

Corke, P., Sikka, P., and Roberts, J. (2000). Height estimation for an autonomous helicopter. In *Int. Symposium on Experimental Robotics*, pages 101–110.

Criminisi, A., Reid, I., and Zisserman, A. (2000). Single view metrology. *Int. J. Comput. Vision*, 40(2):123–148.

de Vries, J. S. and Kemp, R. A. (1994). Results with a multispectral autonomous wildfire detection system. In *Proc. SPIE Infrared Technology XX*, volume 2269, pages 18–28.

Dedeoglu, G. and Sukhatme, G. (2000). Landmark-based matching algorithm for cooperative mapping by autonomous robots. In *Proc. of the Intl. Symposium on Distributed Autonomous Robotic Systems (DARS)*, pages 251–260.

DeGroot, M. and Mortera, J. (1991). Optimal linear opinion pools. *Management Science*, 37(5):546–558.

Del Cerro, J., Barrientos, A., Campoy, P., and García, P. (2002). An autonomous helicopter guided by computer vision for inspection of overhead power cable. In *Proc. of the Workshop WS6 Aerial Robotics*, pages 69–78. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS 2002.

Dempster, A. P. (1968). A generalization of bayesian inference. *Journal of the Royal Statistical Society. Series B (Methodological)*, 30(2):205–247.

Den Breejen, E., Breuers, M., Cremer, F., Kemp, R., Roos, M., Schutte, K., and De Vries, J. (1998). Autonomous forest fire detection. In *Proc. 3rd Int. Conf. on Forest Fire Research*, pages 2003–2012.

Dias, M. and Stentz, A. (2000). A free market architecture for distributed control of a multirobot system. In *Proc. 6th Int. Conf. on Intelligent Autonomous Systems*, pages 115–122.

Dickmanns, E. and Schell, F. (1992). Autonomous landing of airplanes using dynamic machine vision. In *Proc. of the IEEE Workshop Applications of Computer Vision*, pages 172–179.

Dierre, D., Hoff, H., and Bouchet, M. (1999). RAPSODI: Rapid smoke detection and forest fire control. In *Int. Symposium on Forest Fire: Needs and Innovations*, pages 415–419.

Doucet, A., de Freitas, N., and Gordon, N., editors (2001). *Sequential Monte Carlo Methods in Practice*. Springer-Verlag.

Dudek, G., Jenkin, M., and Milios, E. (2002). *Robot teams: from diversity to polymorphism*, chapter A taxonomy of multirobot systems, pages 3–22. A K Peters.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1993). A taxonomy for swarm robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 441–447.

Dudek, G., Jenkin, M., Milios, E., and Wilkes, D. (1998). Topological exploration with multiple robots. In *Proc. of the Intl. Symp. on Robotics with Applications (ISORA)*.

Eck, C., Chapuis, J., and Geering, H. (2001). Inexpensive autopilots for small unmanned helicopters. In *Proc. of the Micro and Mini Aerial Vehicles Conference, MAV2001*.

Elfes, A. (1989). *Occupancy grids: a probabilistic framework for robot perception and navigation*. PhD thesis, Carnegie Mellon University.

Fagg, A., Lewis, M., Montgomery, J., and Bekey, G. A. (1993). The USC autonomous flying vehicle: An experiment in real-time behaviour-based control. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1173–1180.

Farnebäck, G. and Nordberg, K. (2002). Motion detection in the WITAS project. In *Proc. SSAB02 Symposium on Image Analysis*, pages 99–102, Lund. SSAB.

Faugeras, O. (1993). *Three-dimensional computer vision: a geometric viewpoint*. The MIT Press, Cambridge, MA, USA.

Faugeras, O. and Luong, Q.-T. (2001). *The Geometry of Multiple Images: The laws that govern the formation of multiple images of a scene and some of their applications*. The MIT Press.

Feddema, J., Lewis, C., and Schoenwald, D. (2002). Decentralized control of cooperative robotic vehicles: theory and application. *IEEE Transactions on Robotics and Automation*, 18(5):852–864.

Feder, H. J. S., Leonard, J. J., and Smith, C. M. (1999). Adaptive Mobile Robot Navigation and Mapping. *The International Journal of Robotics Research*, 18(7):650–668.

Fenwick, J., Newman, P., and Leonard, J. (2002). Cooperative concurrent mapping and localization. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1810–1817.

Ferruz, J. and Ollero, A. (2000). Real-time feature matching in image sequences for non-structured environments. Applications to vehicle guidance. *Journal of Intelligent and Robotic Systems*, 28:85–123.

Fierro, R., Das, A., Spletzer, J., Esposito, J., Kumar, V., Ostrowski, J., Pappas, G., Taylor, C., Hur, Y., Alur, R., Lee, I., Grudic, G., and Southall, B. (2002). A Framework and Architecture for Multi-Robot Coordination. *The International Journal of Robotics Research*, 21(10-11):977–995.

Finin, T., Labrou, Y., and Mayfield, J. (1997). *Software Agents*, chapter KQML as an agent communication language. MIT Press.

Fleet, D. (1992). *Measurement of Image Velocity*. Kluwer Academic Press, Norwell, MA.

Flint, M., Fernandez, E., and Polycarpou, M. (2004). Efficient bayesian methods for updating and storing uncertain search information for UAVs. In *Proc. of the IEEE Conference on Decision and Control*, pages 1093–1098.

Forssén, P. (2004). *Low and Medium Level Vision using Channel Representations*. PhD thesis, Linköping University. Thesis No. 858.

Forssén, P. and Granlund, G. (2003). Robust multi-scale extraction of blob features. In *Proc. of the 13th Scandinavian Conference on Image Analysis*, LNCS 2749, pages 11–18, Gothenburg, Sweden.

Forssén, P. and Moe, A. (2005). View matching with blob features. In *2nd Canadian Conference on Robot Vision*, pages 228–235, Victoria, BC, Canada. IEEE Computer Society.

Fox, D., Thrun, S., Burgard, W., and Dellaert, F. (2001). *Sequential Monte Carlo Methods in Practice*, chapter Particle Filters for Mobile Robot Localization, pages 401–428. Springer-Verlag.

Fukao, T., Fujitani, K., and Kanade, T. (2003). Image-based tracking control of a blimp. In *Proc. 42nd IEEE Conference on Decision and Control*, volume 5, pages 5414–5419.

Gancet, J. (2005). *Systèmes multi-robots aériens : architecture pour la planification, la supervision et la coordination*. PhD thesis, Laboratoire d'Analyse et d'Architecture des Systèmes (CNRS).

Gancet, J., Hattenberger, G., Alami, R., and Lacroix, S. (2005). Task planning and control for a multi-UAV system: architecture and algorithms. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1017–1022.

Gancet, J. and Lacroix, S. (2004). Embedding heterogeneous levels of decisional autonomy in multi-robots systems. In *Proc. of the 7th International symposium on Distributed Autonomous Robot Systems*.

Garcia-Pardo, P. J., Sukhatme, G. S., and Montgomery, J. F. (2001). Towards vision-based safe landing for an autonomous helicopter. *Robotics and Autonomous Systems*, 38(1):19–29.

Genest, C. and Zidek, J. (1986). Combining probability distributions: a critique and an annotated bibliography. *Statistical Science*, 1(1):114–135.

Gerkey, B., Vaughan, R., and Howard, A. (2003). The Player/Stage Project: Tools for multi-robot and distributed sensor systems. In *Proc. of the International Conference on Advanced Robotics (ICAR).*, pages 317–323.

Ghosal, S. (1999). A review of consistency and convergence rates of posterior distribution. In *Proc. of Varanashi Symposium in Bayesian Inference*.

Giulietti, F., Pollini, L., and Innocenti, M. (2000). Autonomous formation flight. *IEEE Control Systems Magazine*, pages 34–44.

Golub, G. and Loan, C. V. (1996). *Matrix computations*. The John Hopkins University Press, 3rd edition.

Gómez Rodríguez, F., Pascual Peña, S., Arrue, B., and Ollero, A. (2002). Smoke detection using image processing. In *Proc. IV Intl. Congress on Forest Fire Research ICFFR*.

Gonzalo, J. (1998). Fuego: A low cost service for fire detection. In *Proc. 3rd Int. Conf. on Forest Fire Research*, page 2029.

Gordon, N., D.J. Salmond, D., and Smith, A. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107–113.

Grabowski, R., Navarro-Serment, L., Paredis, C., and Khosla, P. (2000). Heterogeneous teams of modular robots for mapping and exploration. *Autonomous Robots*, 8(3):293–308.

Granlund, G., Nordberg, K., Wiklund, J., Doherty, P., Skarman, E., and Sandewall, E. (2000). WITAS: An intelligent autonomous aircraft using active vision. In *Proc. of the UAV 2000 International Technical Conference and Exhibition*, Paris, France. Euro UVS.

Grime, S. and Durrant-Whyte, H. F. (1994). Data fusion in decentralized sensor networks. *Control Engineering Practice*, 2(5):849–863.

Grocholsky, B. (2002). *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, Australian Centre For Field Robotics, University of Sidney.

Grocholsky, B., Makarenko, A., and Durrant-Whyte, H. (2003). Information-theoretic coordinated control of multiple sensor platforms. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1521–1526.

Gunatilaka, A. and Baertlein, B. (2001). Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):577–589.

Hall, J. and Pachter, M. (1999). Formation maneuvers in three dimensions. Technical report, Air Force Institute of Technology.

Haralick, R. and Shapiro, L. (1992). *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.

Harris, C. and Stephens, M. (1988). A combined corner and edge detection. In *Proc. of The Fourth Alvey Vision Conference*, pages 147–151.

Hartley, R. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition.

How, J., King, E., and Kuwata, Y. (2004). Flight demonstrations of cooperative control for UAV teams. In *Proc. of the 3 Annual Unmanned Unlimited Tech. Conf. and Workshop.*

Howard, A., Parker, L., and Sukhatme, G. (2006). Experiments with a Large Heterogeneous Mobile Robot Team: Exploration, Mapping, Deployment and Detection. *The International Journal of Robotics Research*, 25(5-6):431–447.

Hrabar, S. and Sukhatme, G. (2003). Omnidirectional vision for an autonomous helicopter. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 1, pages 558–563.

Hrabar, S., Sukhatme, G., Corke, P., Usher, K., and Roberts, J. (2005). Combined optic-flow and stereo-based navigation of urban canyons for a UAV. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3309–3316.

Huang, W. and Beevers, K. (2005). Topological map merging. *The International Journal of Robotics Research*, 24(8):601–613.

Hudson, R. D. (1969). *Infrared system engineering.* Wiley Series in Pure and Appleid Optics, New York: Wiley, 1969.

Hygounenc, E., Jung, I.-K., Soueres, P., and Lacroix, S. (2004). The Autonomous Blimp Project of LAAS-CNRS: Achievements in Flight Control and Terrain Mapping. *The International Journal of Robotics Research*, 23(4-5):473–511.

Ihler, A., J. W. Fisher, I., Moses, R. L., and Willsky, A. S. (2004). Nonparametric belief propagation for self-calibration in sensor networks. In *IPSN '04: Proc. of the third international symposium on Information processing in sensor networks*, pages 225–233, New York, NY, USA. ACM Press.

Jang, J.-R. (1993). ANFIS: Adaptive-Network-based Fuzzy Inference Systems. *IEEE Transactions on Systems, Man and Cybernetics*, 23(3):665–685.

Jazwinski, A. (1970). *Stochastic Processes and Filtering Theory.* Academic Press, New York.

Johansson, B. and Farnebäck, G. (2002). A theoretical comparison of different orientation tensors. In *Proc. SSAB02 Symposium on Image Analysis*, pages 69–73, Lund. SSAB.

Johnson, A., Montgomery, J., and Matthies, L. (2005). Vision guided landing of an autonomous helicopter in hazardous terrain. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3966–3971.

Julier, S. and Uhlmann, J. (1997a). A new extension of the kalman filter to nonlinear systems. In *Proc. of the 11th Int. Symp. on Aerospace/Defence Sensing, Simulation and Controls.*

Julier, S. and Uhlmann, J. (1997b). A non-divergent estimation algorithm in the presence of unknown correlations. In *Proc.of the American Control Conference*, volume 4, pages 2369–2373.

Jung, I.-K. and Lacroix, S. (2001). A robust interest points matching algorithm. In *Proc. of the Eighth IEEE International Conference on Computer Vision*, volume 2, pages 538–543.

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

Kelhä, V., Rauste, Y., Häme, T., Sephton, T., Buongiorno, A., Frauenberger, O., Soini, K., Venäläinen, A., San-Miguel-Ayanz, J., and Vainio, T. (2003). Combining AVHRR and ATSR satellite sensor data for operational boreal forest fire detection. *International Journal of Remote Sensing*, 24(8):1691–1708.

Kim, J. and Sukkarieh, S. (2004). Autonomous airborne navigation in unknown terrain environments. *IEEE Transactions on Aerospace and Electronic Systems*, 40(3):1031–1045.

Kitano, H., Asada, M., Noda, I., and Matsubara, H. (1998). RoboCup: robot world cup. *IEEE Robotics and Automation Magazine*, 5(3):30–36.

Kjaerulff, U. (1992). A computational scheme for reasoning in dynamic probabilistic networks. In *Proc. of the Eighth Conference on Uncertainty in Artificial Intelligence*, pages 121–129.

Konolige, K., Fox, D., Limketkai, B., Ko, J., and Stewart, B. (2003). Map merging for distributed robot navigation. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 212–217.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86.

Kurazume, R. and Nagata, S. (1994). Cooperative positioning with multiple robots. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1250–1257.

Lacroix, S., Jung, I., Soueres, P., Hygounenc, E., and Berry, J. (2002). The autonomous blimp project of LAAS/CNRS - current status and research challenges. In *Proc. of the International Conference on Intelligent Robots and Systems, IROS, Workshop WS6 Aerial Robotics*, pages 35–42. IEEE/RSJ.

Lemaire, T., Lacroix, S., and Solà, J. (2005). A practical 3D bearing only SLAM algorithm. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2449–2454.

Liggins II, M., Chong, C.-Y., Kadar, I., Alford, M., Vannicola, V., and Thomopoulos, S. (1997). Distributed fusion architectures and algorithms for target tracking. *Proceedings of the IEEE*, 85(1):95–107.

Lindeberg, T. (1994). *Scale-space Theory in Computer Vision*. Kluwer Academic Publishers. ISBN 0792394186.

Ling, L., Ridley, M., Kim, J.-H., Nettleton, E., and Sukkarieh, S. (2003). Six DoF Decentralised SLAM. In *Proc. of the Australasian Conference on Robotics and Automation*.

Ludington, B., Johnson, E., and Vachtsevanos, G. (2006). Augmenting UAV autonomy. *IEEE Robotics and Automation Magazine*, 13(3):63–71.

Lum, C., Rysdyk, R., and Pongpunwattana, A. (2006). Occupancy based map searching using heterogeneous teams of autonomous vehicles. In *Proc. of the AIAA Guidance, Navigation and Control Conference*.

Madhavan, R., Fregene, K., and Parker, L. (2004). Distributed cooperative outdoor multirobot localization and mapping. *Autonomous Robots*, 17:23–39.

Mahony, R. and Hamel, T. (2005). Image-based visual servo control of aerial robotic systems using linear image features. *IEEE Transactions on Robotics*, 21(2):227–239.

Martínez de Dios, J., André, J., J.C.Gonçalves, Arrue, B., Ollero, A., and Viegas, D. (2006a). Laboratory fire spread analysis using visual and infrared images. *Intl. Journal of Wildland Fire*, 15:175–186.

Martínez de Dios, J., Arrue, B., Merino, L., Ollero, A., and Gómez-Rodríguez, F. (2007a). Computer vision techniques for forest fire perception. *Image and Vision Computing*.

Martínez de Dios, J., Merino, L., Caballero, F., Ollero, A., and Viegas, D. X. (2006b). Experimental results of automatic fire detection and monitoring with UAVs. In *Proc. of the V Intl. Congress on Forest Fire Research ICFFR*.

Martínez de Dios, J., Merino, L., Ollero, A., and Viegas, D. (2007b). *Multiple Heterogeneous Unmanned Aerial Vehicles*, chapter Multi-UAV Experiments: Application to forest fires. Springer Tracks on Advanced Robotics. Springer-Verlag.

Martínez de Dios, J. and Ollero, A. (2004). A multiresolution threshold selection method based on training. *Lecture Notes in Computer Science*, 3211:90–97.

Mataric, M. (1992). Minimizing complexity in controlling a mobile robot population. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 830–835.

McLain, T. (2000). Coordinated control of unmanned air vehicles. In *Proc. of the 2000 American Control Conference*.

McLean, G.F.; Kotturi, D. (1995). Vanishing point detection by line clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1090–1095.

Meingast, M., Geyer, C., and Sastry, S. (2004). Vision based terrain recovery for landing unmanned aerial vehicles. In *IEEE Conference on Decision and Control*, pages 1670–1675.

Mejías, L., Saripalli, S., Campoy, P., and Sukhatme, G. S. (2006). Visual servoing of an autonomous helicopter in urban areas using feature tracking. *Journal of Field Robotics*, 23(3-4):185–199.

Merino, L., Caballero, F., Ferruz, J., Wiklund, J., and Ollero, A. (2007a). *Multiple Heterogeneous Unmanned Aerial Vehicles*, chapter Multi-UAV Cooperative perception techniques. Springer Tracks on Advanced Robotics. Springer-Verlag.

Merino, L., Caballero, F., Forssén, P., Wiklund, J., Ferruz, J., Martínez de Dios, J., Moe, A., Nordberg, K., and Ollero, A. (2007b). *Advances in Unmanned Aerial Vehicles. State of the art and the road to autonomy*, chapter Single and multi-UAV relative position estimation based on natural landmarks. Springer Tracks on Advanced Robotics. Springer-Verlag.

Merino, L., Caballero, F., Martínez de Dios, J., Ferruz, J., and Ollero, A. (2006a). A cooperative perception system for multiple UAVs: Application to automatic detection of forest fires. *Journal of Field Robotics*, 23(3-4):165–184.

Merino, L., Caballero, F., Martínez de Dios, J., and Ollero, A. (2005). Cooperative fire detection using unmanned aerial vehicles. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1896–1901, Barcelona, Spain.

Merino, L., Caballero, F., and Ollero, A. (2006b). Applications of networked unmanned aerial vehicles to cooperative fire detection using grid-based data fusion techniques. In *Proceedings of the 8th International IFAC Symposium on Robot Control, SYROCO*. In press.

Merino, L., Martínez Blázquez, R., and Ollero, A. (2004). Cooperative perception for fire monitoring. In *Proc. of the International Symposium on Robotics and Applications ISORA, World Automation Congress*, pages 435–440.

Merino, L. and Ollero, A. (2002a). Computer vision techniques for fire monitoring using aerial images. In *Proc. of the International Conference on Industrial Electronics, Control and Instrumentation*. IEEE.

Merino, L. and Ollero, A. (2002b). Forest fire perception using aerial images in the COMETS project. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 11–22.

Merino, L., Wiklund, J., Caballero, F., Moe, A., Martínez de Dios, J., Forssén, P.-E., Nordberg, K., and Ollero, A. (2006c). Vision-based multi-UAV position estimation. *IEEE Robotics and Automation Magazine*, 13(3):53–62.

Merz, T. (2004). Building a system for autonomous aerial robotics research. In *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*.

Metta, G., Fitzpatrick, P., and Natale, L. (2006). YARP: Yet Another Robot Platform. *International Journal on Advanced Robotics Systems*.

Montgomery, J., Fagg, A., and Bekey, G. (1995). The USC AFV-I: A behavior-based entry in the 1994 International Aerial Robotics Competition. *IEEE Expert*, 10(2):16–22.

Moravec, H. (1989). Certainty grids for sensor fusion in mobile robots. *Sensor Devices and Systems for Robotics*, pages 243–276. Also CMU Robotics Institute 1987 Annual Research Review, 1988, pp. 33-48. Also in AI Magazine v9(2), Summer 1988, pp 61-77.

Morris, P. (1983). An axiomatic approach to expert resolution. *Management Science*, 29(1):24–32.

Musial, M., Brandenburg, U., and Hommel, G. (2000). Cooperative autonomous mission planning and execution for the flying robot marvin. *Intelligent Autonomous Systems*, 6:636–643.

Musso, C., Oudjane, N., and Gland, F. L. (2001). *Sequential Monte Carlo Methods in Practice*, chapter Improving regularised particle filters, pages 248–271. Springer-Verlag.

Mutambara, A. (1998). *Decentralized Estimation and Control for Multisensor Systems*. CRC Press.

Mutambara, A. and Durrant-Whyte, H. (2000). Fully Decentralized Estimation and Control for a Modular Wheeled Mobile Robot. *The International Journal of Robotics Research*, 19(6):582–596.

Nettleton, E., Durrant-Whyte, H., and Sukkarieh, S. (2003). A robust architecture for decentralised data fusion. In *Proc. of the International Conference on Advanced Robotics (ICAR)*.

Nistér, D. (2004). An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770.

Nistér, D., Naroditsky, O., and Bergen, J. (2004). Visual odometry. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 652–659.

Obdrzálek, S. and Matas, J. (2002). Object recognition using local affine frames on distinguished regions. In *13th British Machine Vision Conference*, pages 113–122.

Ollero, A., Alcázar, J., Cuesta, F., López-Pichaco, F., and Nogales, C. (2003). Helicopter teleoperation for aerial monitoring in the COMETS multi-UAV system. In *3rd IARP Workshop on Service, Assistive and Personal Robots*.

Ollero, A., Ferruz, J., Caballero, F., Hurtado, S., and Merino, L. (2004). Motion compensation and object detection for autonomous helicopter visual navigation in the comets system. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 19–24.

Ollero, A., Lacroix, S., Merino, L., Gancet, J., Wiklund, J., Remuss, V., Veiga, I., Gutiérrez, L. G., Viegas, D. X., González, M., Mallet, A., Alami, R., Chatila, R., Hommel, G., Colmenero, F., Arrue, B., Ferruz, J., Martínez de Dios, J., and Caballero, F. (2005). Multiple eyes in the sky: Architecture and perception issues in the COMETS unmanned air vehicles project. *IEEE Robotics and Automation Magazine*, 12(2):46–57.

Ollero, A. and Maza, I., editors (2007). *Multiple Heterogeneous Unmanned Aerial Vehicles*, volume 37 of *Springer Tracks on Advanced Robotics*. Springer-Verlag.

Ollero, A. and Merino, L. (2004). Control and perception techniques for aerial robotics. *Annual Reviews in Control*, (28):167–178. Elsevier (Francia).

Ong, L.-L., Ridley, M., Upcroft, B., Kumar, S., Bailey, T., Sukkarieh, S., and Durrant-Whyte, H. (2005a). A comparison of probabilistic representations for decentralised data fusion. In *Proc. of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing Conference, ISSNIP*, pages 187–192.

Ong, L.-L., Upcroft, B., Ridley, M., Bailey, T., Sukkarieh, S., and Durrant-Whyte, H. (2005b). Decentralised data fusion with particles. Australasian Conference on Robotics and Automation.

Ousingsawat, J. and Campbell, M. (2004). Multiple vehicle team tasking for cooperative estimation.

Pagac, D., Nebot, E., and Durrant-Whyte, H. (1998). An evidential approach to map-building for autonomous vehicles. *IEEE Transactions on Robotics and Automation*, 14(4):623–629.

Pagello, E., DAngelo, A., and Menegatti, E. (2006). Cooperation issues and distributed sensing for multirobot systems. *Proceedings of the IEEE*, 94(7):1370–1383.

Pahliani, A. and Lima, P. (2007). Cooperative opinion pool: A new method for sensor fusion by a robot team. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*.

Papadopoulo, T. and Lourakis, M. (2000). *Computer Vision - ECCV 2000*, chapter Estimating the Jacobian of the Singular Value Decomposition: Theory and Applications, pages 554–570. Lecture Notes in Computer Science. Springer Berlin / Heidelberg.

Parker, L. (1998). ALLIANCE: an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240.

Parker, L. (2000). Current state of the art in distributed autonomous mobile robots. In *Proc. of the Int. Symp. on Distributed Autonomous Robotic Systems*, pages 3–12.

Parker, L. (2002). Distributed algorithms for multi-robot observation of multiple moving targets. *Autonomous Robots*, (12):231–255.

Pastora, E., Águeda, A., Andrade-Cetto, J., noz, M. M., Pérez, Y., and Planas, E. (2006). Computing the rate of spread of linear flame fronts by thermal image processing. *Fire Safety Journal*, 41:569–579.

Proctor, A. and Johnson, E. (2005). Vision-only approach and landing. In *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Proctor, A., Johnson, E., and Apker, T. (2006). Vision-only control and guidance for aircraft. *Journal of Field Robotics*, 23(10):863–890.

Rauste, Y. (1996). Forest fire detection with satellites for forest fire control. In *Proc. of the XVIII Congress of ISPRS*, volume 31 of *International Archives of Photogrammetry and Remote Sensing*, pages 584–588.

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

Rekleitis, I., Dudek, G., and Milios, E. (2003). Probabilistic cooperative localization and mapping in practice. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 1907–1912.

Remuß, V. and Musial, M. (2004). BBCS - efficient communication system for connection of low and high-performance systems. In *Proc. of the First European Micro Air Vehicle Conference (EMAV 2004)*.

Remuß, V., Musial, M., and Brandenburg, U. (2004). BBCS - robust communication system for distributed systems. In *Proc. of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR 2004)*.

Remuß, V., Musial, M., and Hommel, G. (2002). Marvin - an autonomous flying robot-bases on mass market. In *International Conference on Intelligent Robots and Systems, IROS. Proc. of the Workshop WS6 Aerial Robotics*, pages 23–28. IEEE/RSJ.

Rocha, R., Dias, J., and Carvalho, A. (2005). Cooperative multi-robot systems a study of vision-based 3-d mapping using information theory. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 384–389.

Rosencrantz, M., Gordon, G., and Thrun, S. (2003). Decentralized sensor fusion with distributed particle filters. In *Proc. Conf. Uncertainty in Artificial Intelligence*.

Roumeliotis, S. and Bekey, G. (2000). Collective localization: a distributed kalman filter approach to localization of groups of mobile robots. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 2958–2964.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Prentice Hall, second edition.

San Miguel Ayanz, J., Ravail, N., Kelha, V., and Ollero, A. (2005). Active fire detection for fire emergency management: Potential and limitations for the operational use of remote sensing. *Natural Hazards*, 35(3):361–376.

Saptharishi, M., Oliver, C. S., Diehl, C. P., Bhat, K., Dolan, J., Trebi-Ollennu, A., and Khosla, P. (2002). Distributed surveillance and reconnaissance using multiple autonomous atvs: Cyber-scout. *IEEE Transactions on Robotics and Automation*, 18(5):826 – 836.

Saripalli, S., Montgomery, J., and Sukhatme, G. (2003). Visually guided landing of an unmanned aerial vehicle. *IEEE Transactions on Robotics and Automation*, 19(3):371–380.

Saripalli, S., Naffin, D., and Sukhatme, G. (2002). Autonomous flying vehicle research at the University of Southern California. In *Proc. of the First International Workshop on Multi-Robot Systems*.

Saripalli, S. and Sukhatme, G. (2003). Landing on a mobile target using an autonomous helicopter. In *Proc. of the International Conference on Field and Service Robotics, FSR*.

Schmitt, T., Hanek, R., Beetz, M., Buck, S., and Radig, B. (2002). Cooperative probabilistic state estimation for vision-based autonomous mobile robots. *IEEE Transactions on Robotics and Automation*, 18(5):670–684.

Schumacher, C. and Singh, S. (2000). Nonlinear control of multiple UAVs in close-coupled formation flight. In *AIAA Guidance, Navigation, and Control Conference*.

Semple, J. and Kneebone, G. (1952). *Algebraic Projective Geometry*. Oxford University Press. ISBN 0-19-853172-9.

Shafer, G. (1976). *A mathematical theory of evidence*. Princetion University Press, Princetion, NJ.

Shakernia, O., Vidal, R., Sharp, C., Ma, Y., and Sastry, S. (2002). Multiple view motion estimation and control for landing an aerial vehicle. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2793–2798. IEEE.

Sheng, X., Hu, Y.-H., and Ramanathan, P. (2005). Distributed particle filter with GMM approximation for multiple targets localization and tracking in wireless sensor network. In *Proc. of the 4th international symposium on Information processing in sensor networks IPSN*, pages 181–188.

Shi, J. and Tomasi, C. (1994). Good features to track. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 593–600.

Shuo, W. and Min, T. (2000). Multi-robot cooperation and data fusion in map-building. In *Proc. of the 3rd World Congress on Intelligent Control and Automation*, volume 2, pages 1261–1265.

Simmons, R., Apfelbaum, D., Burgard, W., Fox, M., an Moors, D., Thrun, S., and Younes, H. (2000). Coordination for multi-robot exploration and mapping. In *Proc. of the AAAI National Conference on Artificial Intelligence*.

Simmons, R., Smith, T., Dias, M., Goldberg, D., Hershberger, D., Stentz, A., and Zlot, R. (2002). A layered architecture for coordination of mobile robots. In *Multi-Robot Systems: From Swarms to Intelligent Automata, Proceedings from the 2002 NRL Workshop on Multi-Robot Systems*. Kluwer Academic Publishers.

Singh, K. and Fujimura, K. (1993). Map making by cooperating mobile robots. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 254–259.

Smallwood, R. and Sondik, E. (1973). The optimal control of partially observable markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088.

Solà, J., Devy, M., Monin, A., and Lemaire, T. (2005). Undelayed initialization in bearing only SLAM. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2499–2504.

Spletzer, J. and Taylor, C. (2003a). A bounded uncertainty approach to multi-robot localization. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1258–1265.

Spletzer, J. and Taylor, C. (2003b). Dynamic sensor planning and control for optimally tracking targets. *International Journal of Robotics Research*, 22:7–20.

Stancliff, S. and Nechyba, M. (2000). Learning to fly: Modeling human control strategies in an aerial vehicle. In *Proc. of the FCRAR*.

Stoeter, S., Rybski, P., Stubbs, K., McMillen, C., Gini, M., Hougen, D., and Papanikolopoulos, N. (2002). A robot team for surveillance tasks: Design and architecture. *Robotics and Autonomous Systems*, 40(2–3):173–183.

Stone, L. D., Corwin, T. L., and Barlow, C. A. (1999). *Bayesian Multiple Target Tracking*. Artech House, Inc., Norwood, MA, USA.

Stroupe, A., Martin, M. C., and Balch, T. (2001). Distributed sensor fusion for object position estimation by multi-robot systems. In *Proc. of the IEEE International Conference on Robotics and Automation*.

Stroupe, A., Ravichandran, R., and Balch, T. (2004). Value-based action selection for exploration and dynamic target observation with robot teams. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 4, pages 4190–4197.

Sugeno, M., Griffin, M., and Bastia, A. (1993). Fuzzy hierarchical control of an unmanned helicopter. In *Proc. of the 17th IFSA World Congress*, pages 179–182.

Sukhatme, G., Montgomery, J., and Vaughan, R. (2001). *Robot Teams: From Diversity to Polymorphism*, chapter Experiments with Aerial-Ground Robots. AK Peters.

Sukkarieh, S., Nettleton, E., Kim, J.-H., Ridley, M., Goktogan, A., and Durrant-Whyte, H. (2003). The ANSER Project: Data Fusion Across Multiple Uninhabited Air Vehicles. *The International Journal of Robotics Research*, 22(7-8):505–539.

Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics*. The MIT Press.

Thrun, S., Liu, Y., Koller, D., Ng, A., Ghahramani, Z., and Durrant-Whyte, H. (2004). Simultaneous Localization and Mapping with Sparse Extended Information Filters. *The International Journal of Robotics Research*, 23(7-8):693–716.

Triggs, B. (1998). Autocalibration from planar scenes. In *Proc. of the 5th European Conference on Computer Vision, ECCV*, volume 1, pages 89–105, London, UK. Springer-Verlag.

Tsai, R. (1987). A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation [legacy, pre - 1988]*, 3(4):323–344.

Tuytelaars, T. and Gool, L. V. (2004). Matching widely separated views based on affinely invariant regions. *Int. Journal of Computer Vision*, 59:61–85.

Ueyama, T. and Fukuda, T. (1993). Knowledge adquisition and distributed decision making. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 167–172.

Upcroft, B., Ong, L.-L., Kumar, S., Ridley, M., Bailey, T., Sukkarieh, S., and Durrant-Whyte, H. (2005). Rich probabilistic representations for bearing only decentralized data fusion. In *Proc. of the International Conference on Information Fusion*, pages 1054–1061.

Utete, S. and Durrant-Whyte, H. (1994). Reliability in decentralised data fusion networks. In *Proc. of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pages 215–221.

Utkin, A., Fernandes, A., oes A.V. Lavrov, F. S., and Vilar, R. (2003). Feasibility of forest-fire smoke detection using lidar. *International Journal of Wildland Fire*, 12(2):159–166.

Utkin, A., Lavrov, A., Costa, L., oes, F. S., and Vilar, R. (2002). Detection of small forest fires by lidar. *Applied Physics B: Lasers and Optics*, 74(1):77–83.

Utz, H., Sablatnog, S., Enderle, S., and Kraetzschmar, G. (2002). MIRO - middleware for mobile robot applications. *IEEE Transactions on Robotics and Automation*, 18(4):493–497.

Valavanis, K., editor (2007). *Advances in Unmanned Aerial Vehicles. State of the art and the road to autonomy*. Springer Tracks on Advanced Robotics. Springer-Verlag.

Vaughan, R., Sukhatme, G., Mesa-Martinez, J., and Montgomery, J. (2000). Fly spy: Lightweight localization and target tracking for cooperating air and ground robots. In *Proc. of the 5th International Symposium on Distributed Autonomous Robotic Systems*, pages 315–324.

Vidal, R., Sastry, S., Kim, J., Shakernia, O., and Shim, D. (2002a). The Berkeley aerial robot project (BEAR). In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1–10.

Vidal, R., Shakernia, O., Kim, H. J., Shim, D., and Sastry, S. (2002b). Probabilistic pursuit-evasion games: Theory, implementation, and experimental evaluation. *IEEE Transactions on Robotics and Automation*, 18(5):662–669.

Viegas, D. (1998). Forest fire propagation. *Phil. Trans. R. Soc. Lond. A*, 356:2907–2928.

Viegas, D. (1999). Innovations and solutions in fire behaviour prediction issues. In *Proc. of the Delfi International Symposium*, pages 164–173.

Viegas, D. X., Cruz, M., Ribeiro, L., Silva, A., Ollero, A., Arrue, B., Martínez de Dios, J., Gmez-Rodrguez, F., Merino, L., Miranda, A., and Santos, P. (2002). Gestosa fire spread experiments. In *Proc. of the IV International Congress on Forest Fire Research (ICFFR)*, pages 1–13, Coimbra, Portugal.

Weigel, T., Gutmann, J.-S., Dietl, M., Kleiner, A., and Nebel, B. (2002). CS Freiburg: coordinating robots for successful soccer playing. *IEEE Transactions on Robotics and Automation*, 18(5):685–699.

Wimmer, D.-A., Bildstein, M., Well, K., Schlenker, M., Kungl, P., and Kröplin, B.-H. (2002). Research airship "Lotte". Development and operation of controllers for autonomous flight phases. In *Proc. of the Workshop WS6 Aerial Robotics*, pages 55–68. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems - IROS 2002.

Wong, E.-M., Bourgault, F., and Furukawa, T. (2005). Multi-vehicle bayesian search for multiple lost targets. In *Proc. of the IEEE International Conference on Robotics and Automation*, pages 3169–3174.

Wu, A., Johnson, E., and Proctor, A. (2005). Vision-aided inertial navigation for flight control. In *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Yakimenko, O., Kaminer, I., Lentz, W., and Ghyzel, P. (2002). Unmanned aircraft navigation for shipboard landing using infrared vision. *IEEE Transactions on Aerospace and Electronic Systems*, 38(4):1181–1200.

Yamauchi, B. (1998). Frontier-based exploration using multiple robots. In *Proc. of the 2nd Int. Conference on Autonomous Agents*, pages 47–53.

Yanli, Y., Minai, A., and Polycarpou, M. (2005). Evidential map-building approaches for multi-UAV cooperative search. In *Proc. of the American Control Conference*, pages 116–121.

Yu, B., Sycara, K., Giampapa, J., and Owens, S. R. (2004). Uncertain information fusion for force aggregation and classification in airborne sensor networks. In *AAAI-04 Workshop on Sensor Networks*.

Zadeh, L. A. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, (1):3–28.

Zhang, H. and Ostrowski, J. (1999). Visual servoing with dynamics: control of an unmanned blimp. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 1, pages 618–623.

Zhang, Z. (1995). Parameters estimation techniques. a tutorial with application to conic fitting. Technical report, INRIA, France.

Zhang, Z. (1997). Parameters estimation techniques. a tutorial with application to conic fitting. *Image and Vision Computing*, 15(1):59–76.

Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(11):1330–1334.

Zhang, Z. and Hintz, K. (1995). Evolving neural networks for video attitude and hight sensor. In *Proc. of the SPIE International Symposium on Aerospace/Defense Sensing and Control*, volume 2484, pages 383–393.

Zhou, G., Li, C., and Cheng, P. (2005). Unmanned aerial vehicle (UAV) real-time video registration for forest fire monitoring. In *Proc. IEEE Intl. Geoscience and Remote Sensing Symposium IGARSS*, volume 3, pages 25–29.

Zlot, R., Stentz, A., Dias, M., and Thayer, S. (2002). Multi-robot exploration controlled by a market economy. In *Proc. of the IEEE International Conference on Robotics and Automation*, volume 3, pages 3016–3023.