Reliable perception of highly changing environments: implementations for car-to-pedestrian collision avoidance systems

Gwennaël Gate

ED n^o 431 : Information, Communication, Modélisation et Simulation

THESE

pour obtenir le grade de

DOCTEUR DE L'ECOLE NATIONALE SUPERIEURE DES MINES DE PARIS

Spécialité "Informatique temps réel, Robotique, Automatique"

présentée et soutenue publiquement par

Gwennaël GATE

le 3 décembre 2009

Reliable Perception of Highly Changing Environments

Implementations for Car-to-Pedestrian Collision Avoidance Systems

Directeur de thèse : Dr. Fawzi Nashashibi

Jury

M.	Michel DEVY	Rapporteur
M.	Didier AUBERT	Rapporteur
M.	Alberto BROGGI	Examinateur
M.	Christian LAUGIER	Examinateur
M.	Laurent TRASSOUDAINE	Examinateur
M.	Benoist FLEURY	Examinateur
M.	Fawzi NASHASHIBI	Examinateur
M.	Claude LAURGEAU	Examinateur

To A.L, A. and...
To my parents and grand parents.

Abstract

Robots have been given sophisticated "eyes" to make them "see" and understand their environments. These eyes (cameras, ladars, sonars, radars, etc...) collect a huge amount of data that need to be correctly processed to be useful. Processing this information is what a *perception system* is intended to perform.

For almost half a century now, various perception algorithms have been proposed to tackle one or several of the underlying issues that arise when addressing the perception problem. Well known tracking, detection, mapping, localization and classification algorithms can consequently be combined to design complete perception algorithms that work well for a given application in most situations.

The problem is that some real world applications (autonomous driving, etc...) require perception systems that do better than working well in most situations. An autonomous vehicle driving in a crowded urban center would need indeed to be equipped with a perception system that works well in every situation.

This dissertation addresses the specific problem of perception systems reliability when confronted to highly changing dense environment.

First a detailed analysis of the fundamental limitations undermining the performances of existing approaches is given. Then an original approach - based on a unified grid-based formulation of the five perceptual subproblems - is proposed and proves to be capable of solving issues that most existing systems cannot solve.

The relevance of this analysis and the experimental validity of the proposed approach is assessed through an experimental comparison of two fully detailed original perception systems specifically designed for pedestrian detection purposes in urban environments.

Résumé

La plupart des systèmes robotiques sont équipés de capteurs sophistiqués censés leur donner la capacité de "voir" et en conséquence de comprendre l'environnement dans lequel ils évoluent. Cependant, la quantité impressionnante d'information que ces capteurs collectent régulièrement n'est réellement mise à profit que si le robot qui en est doté possède la capacité de les traiter correctement.

Depuis plusieures décennies, une grande variété d'algorithmes de perception a été proposée à cet effet. Il est donc déjà possible d'assembler des algorithmes bien connus de détection, de pistage, de classification, de cartographie et de localisation pour concevoir des systèmes de perception complets capables d'opérer, pour une application donnée, dans la plupart des situations.

Malheureusement, un certain nombre d'applications concrètes exigent des systèmes de perception qui font bien mieux que de fonctionner la plupart du temps. Un véhicule automatique (sans conducteur) évoluant dans un centre ville ne pourra par exemple se satisfaire que d'un système de perception qui fonctionne dans toutes les situations.

Ce mémoire de thèse traite précisément du problème de fiabilité inhérent aux systèmes de perception actuels lorsqu'ils sont confrontés à des environnements complexes et changeants.

Une analyse détaillée des causes de ce manque de fiabilité est d'abord proposée. Nous proposons et décrivons ensuite une approche nouvelle du problème de perception basée sur une formulation unifiée de ses cinq problèmes sous-jacents (détection, pistage, classification, cartographie et localisation). Nous montrons ainsi que cette approche permet de contourner naturellement les difficultés qui bornent les performances de la plupart des systèmes existants.

La pertinence de l'analyse présentée dans ce document ainsi que la validité expérimentale des solutions proposées sont évaluées au travers d'une comparaison concrète entre deux systèmes de perception originaux conçus pour "percevoir" des piétons en environnement urbain.

Acknowledgements

For his priceless technical advice, insights and guidance but also for the constant trust he put on my work, I would like to thank first my advisor Fawzi Nashashibi. His expertise and his daily cheerfulness have both been of great support over these three years.

I would also like to thank Amaury Breheret for his technical contributions to this work and for having accepted to be regularly the "reckless pedestrian" of my experiments.

I am also especially grateful to Professor Claude Laurgeau and Arnaud de la Fortelle who allowed me to pursue my research in their laboratory. Thank you also to Christine Vignaud, Konaly Sar and Christophe Kotfila for their administrative support.

My participation to the national project "LOVe" gave me the opportunity to share insightful discussions with a great number of researchers. Among them, I would especially like to thank Véronique Cherfaoui, Didier Aubert and Benoist Fleury from whom I learned a lot.

These three years at the Robotic center would have certainly been different without the liveliness of its members. I want to thank especially Laure Leroy, Taha Ridene, Alexandre Bargeton, Raoul de Charette, Vincent Meyrueis, Jean-Emmanuel Deschaud, Alexis Paljic, Eric Vecchie, Omar Hamdoun, Fatin Zaklouta, Bogdan Stanciulescu, Keerthi Naravan, Oussama El Hamzaoui, Anne-Sophie Puthon and Sung-Woo Choi, with whom I had the great pleasure to live and work.

For giving me the opportunity to work with him at Carnegie Mellon University some years ago, I want to cheerfully thank Professor Drew Bagnell. His expert guidance and his encouragements have played a major role in my wish to undertake this PhD work.

I am grateful to my parents and my grand parents whose love and pedagogy allowed me to find the way to pursue the exciting studies I was dreaming of.

Finally, I want to thank my wife Anne-Laure for her patience, her support, her encouragements and her love.

Gwennaël Gâté

Contents

Chapter 1

Introduction

Contents

1.1 Mobile Robots, Perception and Reliability

Robots have been used for almost half a century now in a great number of situations. Industrial production, planetary or mines exploration are domains where robots play a decisive role but surprisingly, in far less extreme everyday life environments, robots have still difficulties to find their ways.

The main reason of this is arguably the fact that most everyday life situations are much less predictable than the highly structured environment of a plant or the static landscape of a planet. In other words, mobile robots should be able to cope with highly changing environments to start helping us in our daily duties.

Cleaning, driving, lawn mowing are all examples of everyday life tasks that robots might be able to perform autonomously and safely one day. This is unfortunately conditional to their ability to "perceive" and "understand" the highly dynamical environments related to these tasks.

Over the last fifty years, significant efforts have been carried out to build sensors capable of making observations about the environment and to design autonomous perception system able to process and disambiguate the huge amount of data collected by these sensors.

As a result, a great variety of different approaches have been proposed to solve the different issues that arise when tackling the perception problem. However, most approaches are only intended to work under specific assumptions and are not always scalable to highly changing

outdoor environments. These algorithms are nevertheless commonly used to tackle the challenging environments they were not originally designed to deal with and reasonable results can in practice be obtained.

This is at least the case in most situations where objects in the scene happen to be easy to discriminate and to classify. But unfortunately because of their original limitations, these algorithms will inexorably and regularly face more uncommon situations where they will fail to correctly understand what is going on.

Most perception approaches for highly changing environments are consequently "satisfactory" in many situations but unfortunately not "reliable". These systems are then only usable in applications where a potential failure do not endanger human security.

This lack of reliability of perception algorithms is probably the main reason that explains why we are still driving our cars, washing our knives or mowing our lawns.

The main objective of this dissertation is to address this problem by analyzing first what makes existing perception systems inherently unreliable and by then proposing an original solution to reliable perception of highly changing environments.

1.2 A Specific Application: Collision Avoidance

1.2.1 Motivations

The work presented in this dissertation has been initially motivated by a specific potential application: onboard collision avoidance systems. Detecting pedestrians in urban environments for collision avoidance purposes is indeed a typical example where a high level of reliability is of paramount importance. Besides, the typical complexity of urban environments makes the perception task especially demanding due to the great variety of objects that they contain and because of the severe occlusions that usually undermine the sensor measurements.

Pedestrian perception in urban centers is consequently the specific application to which is confronted the analysis and the contributions proposed in this dissertation.

To fully understand the extent of this dissertation a brief overview of what a vehicle-to-pedestrian collision avoidance is expected to do is given in the next sections as well as the main requirements that its perception subsystem should meet.

1.2.2 Principles

Pedestrian casualties related to collisions with vehicles is still a worrying problem. In 2005, 635 pedestrians ($\simeq 1.75$/day) were killed on french roads and 13,609 were injured ($\simeq 37$/day)[1]. Moreover, 2/3 of these accidents occur in urban areas and 75% of these pedestrians are injured while crossing a street. Studies show also that the great majority of these accidents is due to the driver lack of attention.

Parallel investigations have shown that the death probability of a pedestrian in a collision with a vehicle is highly dependent on its speed and decreases quickly between 60km/h and 40km/h as shown in figure 1.1.

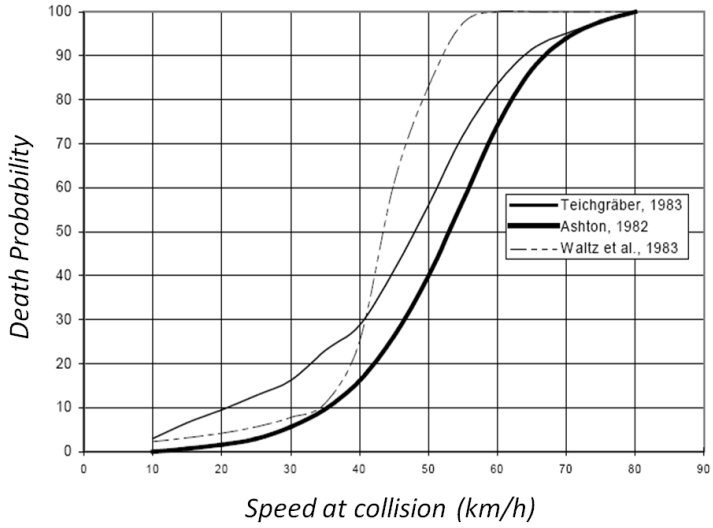


Figure 1.1: The evolution of the probability that a pedestrian dies in a collision with a vehicle of a given speed (from the European Transport Safety Council).

Based on this statistic analysis, it is then possible to imagine a driver assistance system that if not capable of autonomously avoiding a collision with a pedestrian can at least decrease its probability of death.

The idea is simple: when the system detects a pedestrian that is likely to be injured, an alarm is released that makes the driver focus on the situation and initiate the appropriate emergency braking. By shortening the reaction time of the driver, the driver brakes sooner, the collision speed and the probability of death are then decreased. This is schematized in figure 1.2.

It is important to note that despite its name a collision avoidance system does not necessarily avoid a collision but is instead expected to lower its gravity (the term of "pre-crash system" is hence sometimes used).

[1] 2005 Statistics of French Road Safety, Ministry of Transport, June 2006.

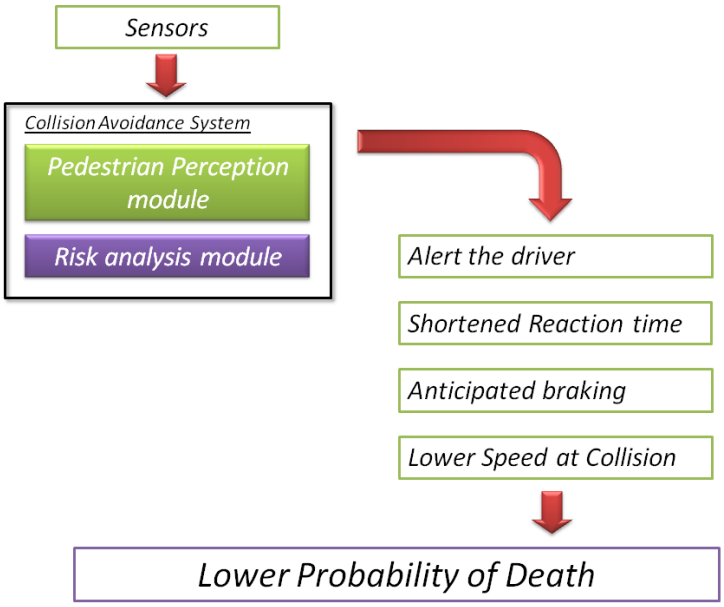


Figure 1.2: Schematic view of a vehicle-to-pedestrian collision avoidance (pre-crash) system.

1.2.3 Requirements

A pedestrian perception module intended to feed such a system should consequently provide a list of all the pedestrians that could be injured by the sensing vehicle in a next future to the risk analysis module. This former module is indeed expected to decide which pedestrian is threatened and is likely to require at least the following features for each detected pedestrian:

- Coordinates of the pedestrian in a frame centered on the vehicle.
- Current velocity of the pedestrian. This estimate can be relative to the speed of the vehicle, but an estimate of the absolute speed is desirable.
- An indicator of the uncertainty related to the information provided above.

The perception system should be able to release these features while simultaneously satisfying three constraints.

Precision

Because releasing a false alarm can potentially provoke unpredictable driver reactions and hazardous situations, a high level of precision is here a critical requirement. The precision of the whole collision avoidance system is naturally dependent on the perception module ability to detect "only" pedestrians. It is interesting to note that missing a pedestrian (sometimes called the *recall* capability of the system) is a less critical requirement as doing nothing is probably safer than doing bad things in this case.

Robustness

Another facet of what usually falls under the general term of "reliability" is what can be called the robustness of a perception system. This specifically refers to the ability of the perception system to work with the same degree of precision in various situations including bad weather conditions, extreme illumination conditions (day/night), crowded areas, etc...

Computational friendship

Finally, because the power of onboard computer units is limited, such a perception module should be very reasonable in terms of computational load and memory requirements.

1.2.4 Collaboration with Automotive Companies

Utilizing this specific application as a test bed for the work presented in this dissertation is also motivated by the fact that the work presented in this dissertation has benefited from the guidance of two major actors of the automotive industry *RENAULT* and *VALEO* through a national research project named *LOVe* which stands for (*Logiciels d'Observation des Vulnérables - Perception Systems for Vulnerable Pedestrians*).

The rich experience provided by these two companies along with the constructive research work performed by the 10 academic contributors of this project made of this specific application the ideal starting point for constructing the more general analysis of autonomous perception in complex environments presented in this dissertation.

1.3 Thesis Statement

This dissertation is intended to meet two objectives.

1. Give a comprehensive analysis of the fundamental limitations that undermine the reliability of existing perception approaches in highly changing environments.

2. Propose a powerful approach that overcomes these limitations and enables a new level of reliability.

This analysis is based on the successive and detailed presentations of two different pedestrian perception algorithms whose performances are compared on a set of real situations especially difficult to handle.

1.4 Outlines of the Dissertation

The presentation of this work is organized in 6 chapters with the following underlying logic.

Chapter 2

A complete overview of *state of the art* perception algorithms is given in Chapter 2 as well as the necessary definitions and mathematical foundations needed to understand the scope of this dissertation. This chapter is not intended to enumerate a set of methods but aims instead at building a taxonomy of existing algorithms based on the specific problems they address (Detection, Tracking, Classification, Mapping, Localization) and on their capacity to handle uncertainties.

Chapter 3

Based on this analysis, a first pedestrian perception system is entirely detailed in chapter 3. This perception algorithm is intended to be an objective example of what can be achieved using existing perception approaches. This system based on the contribution of a ladar and a camera is however innovative in the way uncertainties are managed throughout the process and in its capability to track both pedestrians and groups of people. Extensive online experiments results are presented and the precision of this algorithm is quantitatively assessed to serve as a reference for the analysis given in Chapter 4.

Chapter 4

Based on the system presented in Chapter 3, a detailed analysis is given about the four main issues that no existing approach is able to simultaneously handle. This analysis leads to the identification of a set of specific situations where the algorithm presented in Chapter 3 will inexorably fail as would presumably do most existing approaches. The capabilities of some recent so called *SLAM with DATMO* algorithms are also discussed.

Chapter 5

In Chapter 5, an original grid-based approach is proposed to overcome simultaneously the four limitations identified in Chapter 4. The principles of this approach as well as the mathematical foundations necessary to implement this approach are presented. The general outlines of a grid-based general perception algorithm are also given. This algorithm is described in a general context and does not specifically focus on the pedestrian perception problem.

Chapter 6

To assess the benefit of the approach proposed in Chapter 5, a second pedestrian system is thoroughly detailed in chapter 6. This practical implementation of the theoretical concepts presented in Chapter 5 allows to valid on simulated data some of its important capabilities. Finally, the last sections of this chapter are dedicated to the evaluation of this new pedestrian system on the set of challenging real situations identified in Chapter 4.

Chapter 7

In Chapter 7, a brief summary as well as possible extensions to this work are presented.

Résumé en français du chapitre 2

Nous tentons de fournir au lecteur de ce chapitre une vue générale de l'état de l'art dans le domaine de la perception automatique ainsi que les notions mathématiques et l'explication des termes techniques nécessaires à la lecture de ce mémoire. Ce chapitre à pour principal objectif de présenter et de catégoriser les algorithmes existants en fonction du sous-problème qu'ils tentent de résoudre (détection, pistage, classification, cartographie et localisation) et de leur capacité à prendre en compte l'incertitude des données sur lesquelles ils travaillent ainsi que celle qu'ils génèrent immanquablement.

Chapter 2

Foundations of Autonomous Perception

Contents

2.1 Introduction

This chapter is intended to meet two objectives. First, it aims at defining and clarifying scientific terms that will be used in the rest of this dissertation as *shortcuts* for referring to a very specific concept or problem.

Second, it gives a comprehensive and synthetic overview of state of the art approaches to autonomous perception problems in mobile Robotics. This is indeed a crucial prerequisite for an objective appreciation of the original approaches proposed in chapter 5 and 6 of this document.

2.2 Definitions

It is important to define first the terms of the perception vocabulary in order to fully understand the scope of this dissertation.

2.2.1 Sensing Platform

Any device (static or not) equipped with one or several sensors will be referred to as a *sensing platform*. Although this dissertation is mainly concerned with sensing vehicles applications, the vast majority of the material presented here can be directly used on any other type of mobile devices equipped with sensors. The term *sensing platform* will be used consequently to refer to any of these mobile devices.

Figure 2.1: Different types of sensing platforms that are considered in this dissertation. The two vehicles are *INRIA* and *Mines ParisTech* sensing platforms used as test platform in this thesis work. The bottom left picture show the mobile robot designed by *Willow Garage* and the bottom right picture shows the platform developed by *Aldebaran Robotics*.

2.2.2 Environment and Scene

The *environment* refers to the context within which a sensing platform collects data. Note that a sensing platform will rarely be able to collect data of its whole environment at the same time. That is why, the visible portion of the environment from the sensing platform point of view will be called a *scene*.

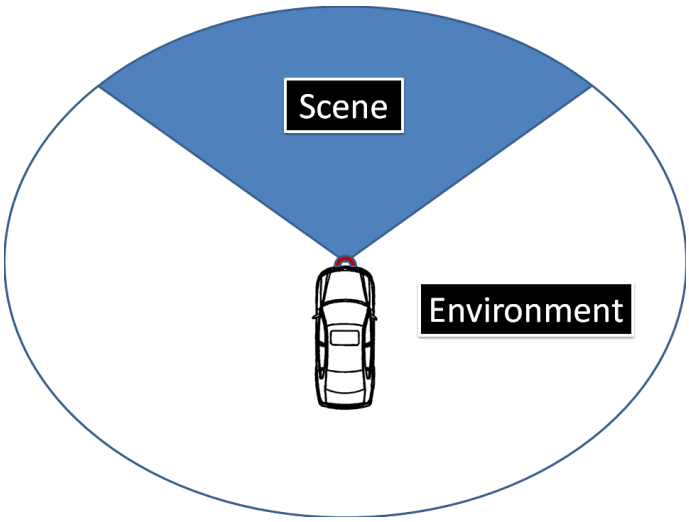


Figure 2.2: The portion of the environment that is visible from a specific sensor is referred to as a scene in this dissertation.

2.2.3 Objects

We consider in this dissertation that the environment can be decomposed into several *objects* (pedestrians, vehicles, walls, sidewalks, tags on a wall, herbs, trees, etc...). It is however important to point out that the nature of what will be referred to as *objects* will vary a lot depending on the application.

A robot intended to gather apples in a tree will probably have to consider every single branch as an object while a robot only intended to navigate safely in a given environment will probably only consider the whole tree as an object.

While taking into consideration that there is not a unique way to decompose an environment into *objects*, this environment theoretical representation will be used throughout this dissertation for simplicity. An example of the same scene being decomposed into *objects* differently depending on applications is given in figure 2.3.

We will also assume that each object in the environment can be described by a vector in a *state space*. In most situations, this space is of course of high dimension and will not be directly used in practice. Low dimension, simplified state spaces will be used instead to represent the knowledge acquired by a perception system about its environment as discussed in the section

2.4 of this chapter. The elements of an object state vector (position, dimensions, velocity, color, etc...) will be referred to as *state parameters* or *object features* in the following.

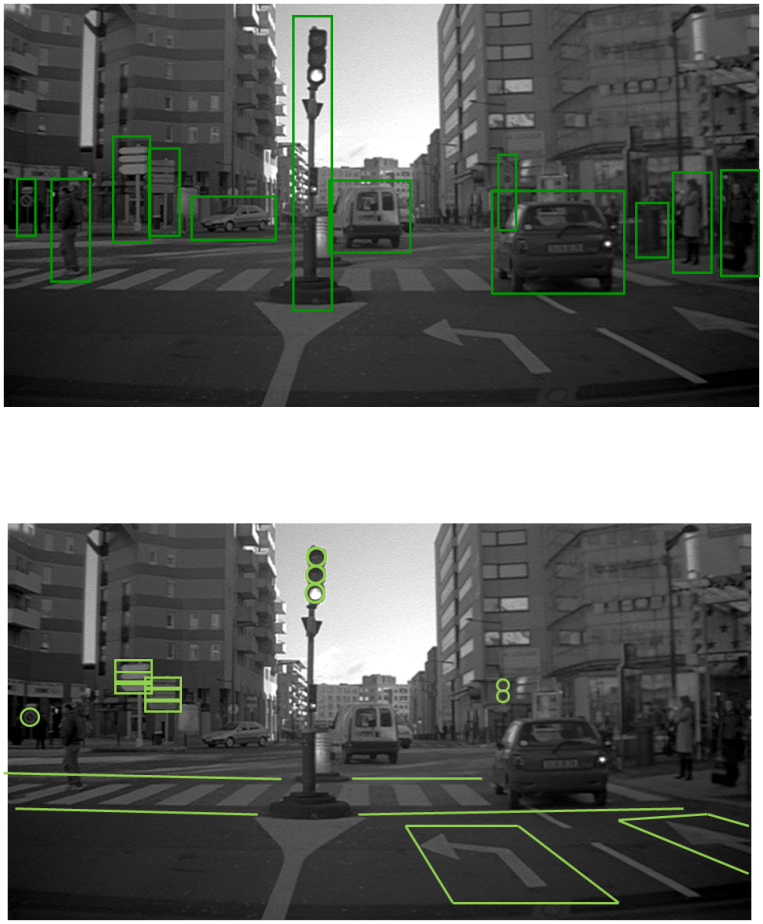


Figure 2.3: On the left: Ideal decomposition of a scene into *objects* for obstacle avoidance applications. On the right: Ideal decomposition of a scene into *objects* for road signs detection applications.

2.2.4 Perception Tasks

Perception is also a term that needs to be defined. In a dictionary, the following definition can be found:

> *Perception is the process of transforming sensations into knowledge about the world. This knowledge is then summarised into an internal representation, which is the result of the perception process.*

In the context of autonomous perception this definition can be refined. It is assumed in this dissertation that *perceiving* a scene can be seen as solving five perceptual tasks: *Detection, Tracking, Classification, Mapping* and *Localization.*

Consequently, solving the perception problem will in principles directly lead to the accomplishment of these five tasks. In the following paragraphs, a brief description of every perceptual task is given.

Detection

Decide which sensors measurements or equivalent entities belong to the same objects in the scene.

Sensors collect regularly information about a scene and send a batch of raw data that is not specific to one object in the scene but instead is a representation of "all" the visible objects in the scene. The *detection problem* (sometimes called the *segmentation problem*) aims at grouping sensors measurements that relate to the same object. This task, simple for humans, can be very difficult for autonomous systems as sensor data do not always provide directly all the relevant data for accurate detection.

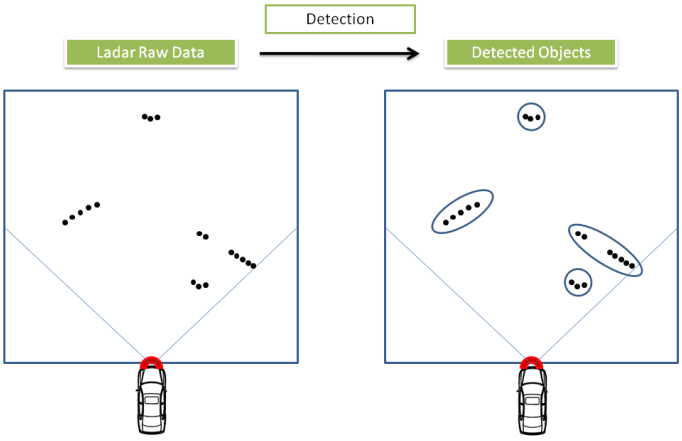


Figure 2.4: Schematic view of a detection process from ladar raw data.

Tracking

Estimate the spatiotemporal trajectory of every "detected objects" or equivalent entities over time.

The need to track objects in the environment comes from the fact that sensors provide data that are both incomplete and noisy. As a consequence, some important features (like object velocities) may not be measured by a sensor and measurements that are collected do not reflect the reality perfectly. By keeping a trace of every previously detected object, it is possible to smooth the knowledge of the system about measured features and to estimate unobserved ones.

Tracking is thus a term that hides two different questions. The first is concerned with the association of previously detected objects with the new sensor observations and is usually referred to as the *data association* or *registration* problem. The second one is concerned with the way the former knowledge about an object is combined with the new data and is usually called the *filtering* problem.

It is important to note that tracking can be used to estimate many different object features and not just its real position and velocity. In that sense, tracking should not necessarily refer to spatiotemporal trajectory estimation. However, because object positions and velocities are usually needed to ensure correct motion prediction and subsequent good data association, the term tracking is in practice always linked to object kinematics estimation in the literature. As a result, the process of estimating the real objects positions and velocities will be referred to as tracking in this dissertation even if it can be seen as a confusion between the name of a problem (object kinematics estimation) and the name of a method to solve it (tracking).

An object that is currently tracked and from which some kinematics features are estimated is usually called a *tracked object* or a *track*.

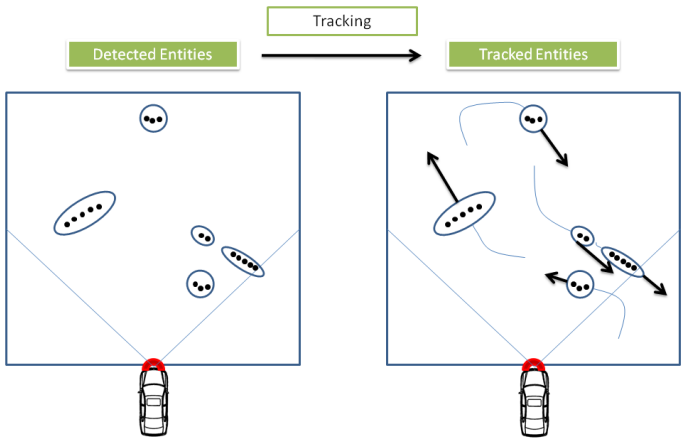


Figure 2.5: Schematic view of a tracking process from ladar raw data.

Classification

Decide what is the "type" of all the objects present in the scene.

Applications usually (but not always) require some autonomous recognition of the objects present in a scene. For example, classification can be mandatory for applications such as cars *Auto-Cruise Control* where only surrounding vehicles should be considered for the speed regulation of the sensing car. Classifying objects is difficult for two reasons.
First, it is a task that heavily depends on the accurate computation of the detection and the tracking tasks as they usually provide useful information for classification (geometry, colour, velocities, etc...).
Second, real objects are rarely similar to any theoretical models. As a consequence the system needs to use criteria that are vague enough to be independent from a particular object within a specific class ("a pedestrian is between 1m and 2m tall") and at the same time specific enough to allow discrimination with objects belonging to other classes.

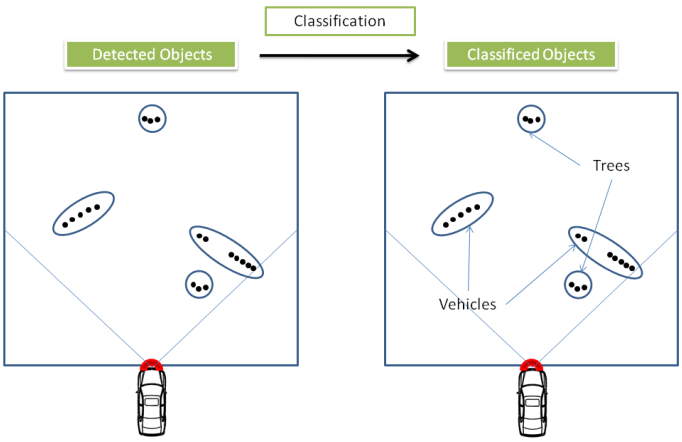


Figure 2.6: Schematic view of a classification process from ladar raw data.

Mapping

Build and refine over time the outlines of every "object" in the scene.

A sensors is usually unable to collect relevant information about its entire surrounding environment directly. At a certain point in time, some objects are out of its reach while some others are partially or totally occluded. In any case, the currently visible outlines of objects might be insufficient to describe the environment correctly. In many application, it is thus desirable to build over time a complete description of object outlines. This description is usually represented as a map as depicted in figure 2.7.

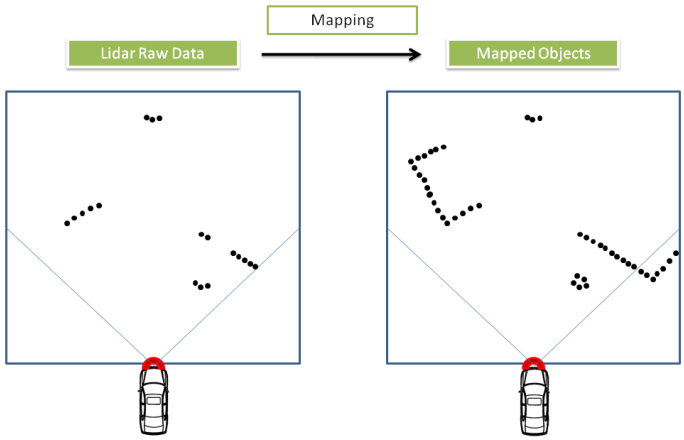


Figure 2.7: Schematic view of a mapping process from ladar raw data.

Localization

Compute an estimate of the robot pose and orientation in the environment.

Localization is a critical issue for any mobile platform that needs to autonomously go from one point in space to another. Localization can be achieved using a lot of approaches. Specific devices like *GPS* receivers, odometers or *IMU* can of course be used to locate the sensing platform. But a very natural way for a mobile platform to quantify its displacement is to look at the induced displacement of the objects present in the scene. The main difficulties lay in the fact that a correct localization is only achieved if the kinematics of the objects present in the scene are known.

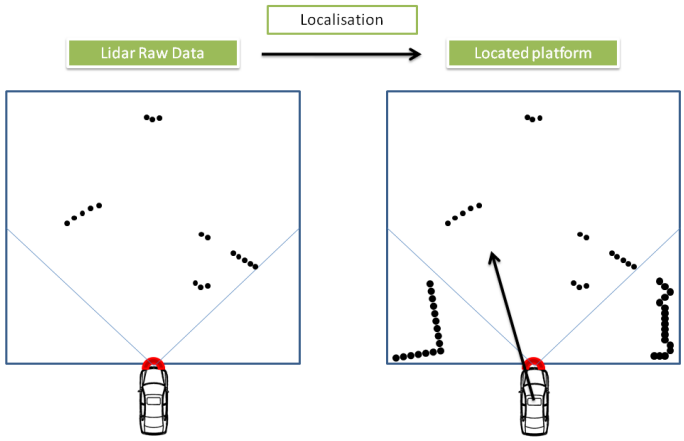


Figure 2.8: Schematic view of a localization process from ladar raw data.

2.2.5 Interdependence of the five perceptual tasks

It is crucial to note that the five perceptual tasks presented in the previous section are the five facets of the same global problem. The computation of any given perceptual task is as a result highly dependent on any other perceptual tasks.

Indeed, by providing kinematics estimates of some entities, an efficient tracking can help the detection process. Conversely, tracking depends on the accurate detection of the entities of interest in the raw data. A good object mapping can help the classification process by giving refined geometrical information. The tracking process is highly dependent on a correct sensing platform localization, etc...

To prove that the five perceptual tasks are all mutually beneficial, a systematic description of the possible interactions that could be used between every task is given.

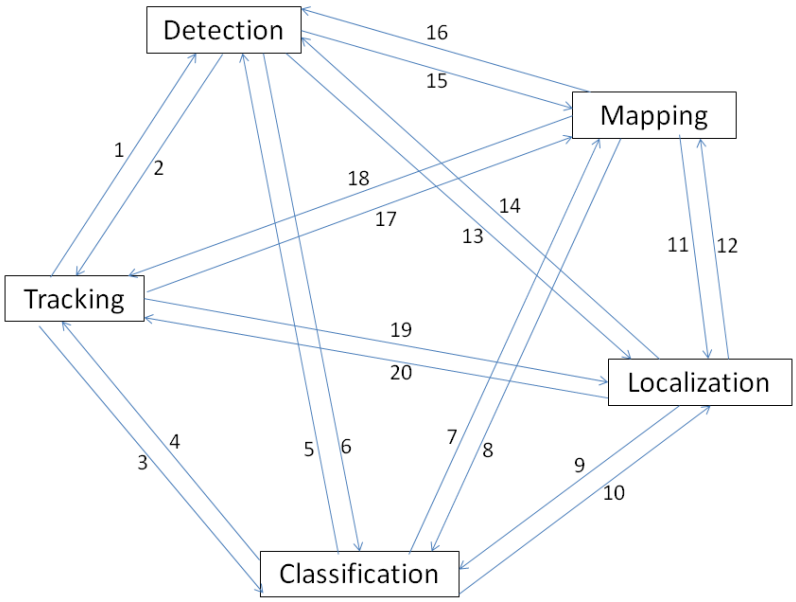


Figure 2.9: The accurate computation of a perceptual task depends in principle on the accurate computation of the remaining perceptual tasks.

Description of the interactions between the perceptual tasks

1. By computing objects kinematical features, tracking can help detection.
2. By grouping sensor observations into (parts of) objects, detection can facilitate tracking.
3. By computing objects kinematical features, tracking can help classification.

4. By indicating the nature of an object, classification can help tracking to use the right motion model for prediction and data association to find realistic correspondences.
5. By indicating the nature of an object previously seen in a specific area, classification can facilitate the correct clustering of sensor measurements.
6. By grouping sensor measurements into objects, detection is almost a prerequisite before classification.
7. By indicating the nature of an object, an algorithm that estimates objects outlines can be oriented.
8. By producing accurate geometrical information about an object, mapping can be very useful to classification.
9. As tracking only deals with already seen objects, a good localization can be necessary to use the sensor observations relevant for the classification of a moving object seen for the first time.
10. By providing the static or moving nature of scene objects, classification can be very useful to localization.
11. By providing a reference to which can be compared new sensor observations, mapping is a prerequisite to environment based localization.
12. As tracking only deals with already seen objects, localization is necessary to incorporate sensor observations that are seen for the first time into the map.
13. By providing detected objects in the environment, detection allows the number of possible trajectories to be decreased (the sensing platform will not go through obstacles) and the localization is thus facilitated.
14. By allowing the correct alignment of sensor observations unrelated to any previously seen objects with the map, localization permits for example the detection of moving objects (being those that fall into previously unoccupied space.).
15. Knowing that two entities in the map belong to the same real object can permit for example to homogenize the map of this object.
16. Conversely, by providing rich geometrical descriptions of entities, the clustering of such entities into relevant objects can be facilitated.
17. By finding the correspondence between the current map and the new measurements, a correct tracking can permit to compute accurate maps of both static and moving objects.
18. By providing rich geometrical description of objects, mapping can enable accurate data association.

19. By solving the data association problem between two scans, tracking is necessary for accurate localization.

20. A good sensing platform localization is necessary to estimate accurate object dynamical features and hence to ensure correct object tracking.

2.2.6 Partial perception systems

Because of the interdependent nature of the five perception tasks, they should in principle be all performed simultaneously. However, the vast majority of existing perception systems make the implicit assumption that some of the perceptual tasks mentioned above can be treated independently or at least sequentially. This assumption is arguably the most commonly made in perception systems design. This even led to the emergence of two distinct scientific communities: the multiple-object tracking (MOT) community that addresses the detection, tracking and classification problems and the simultaneous localization and mapping ($SLAM$) community that aims at solving the mapping and localization problems.
Of course, there are two good reasons for that:

1. Satisfying performances can usually be obtained on specific applications without taking into account full dependencies between the perceptual tasks.

2. In most situations, solving properly the five perceptual tasks simultaneously is impossible in practice as it can be seen as a hard inference problem in highly dimensional continuous spaces.

If systems that solve only some of the five perceptual task can perform well in lot of situations, they tend to be very sensitive to some specific situations where unestimated features are needed to correctly disambiguate the data. Unfortunately, those difficult situations arise frequently in highly changing environments. A complete analysis of these difficult situations is given in chapter 4.

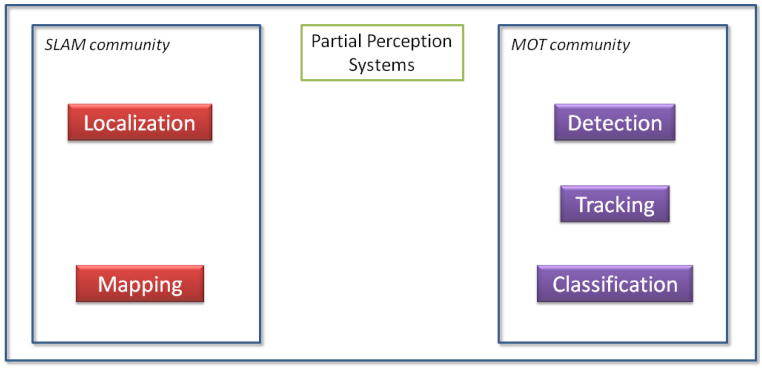


Figure 2.10: Historically, perceptual tasks have been addressed separately by the $SLAM$ and the MOT communities.

2.3 Sensors

Designing a perception system is usually very dependent on the sensor used to collect data for at least two reasons. First, the nature of the data that is collected varies with the type of sensor leading directly to different requirements in the processing of the perceptual tasks. Second, data sent by a sensor is always noisy and as such make the processing of the perceptual tasks even more demanding. Depending on the specification of the sensor, autonomous perception systems will thus have to handle the uncertainties generated by the sensor noisy measurements.

Because sensors have a direct impact on perception algorithms, a brief description of existing sensing technologies is given in this section. This section aims also at underlining the comparative advantages and disadvantages of every technology. This brief analysis will be used in the next chapter to propose a first perception strategy for pedestrian detection in urban environments.

2.3.1 Sensing technologies

Range Sensors

Four types of sensors have the ability to collect range data: *ladar*, *radar*, *sonar* and *stereo-vision based sensors*. Ladars, radars and sonars analyse the response of a scene to signals they emit. Stereo-vision based sensors use one or several cameras to compute (along time when only one camera is used) depth information.

Ladars (laser scanners)

They operate by the emission of pulsated beams of light in the near-infrared frequencies. The range to an object is measured using the pulse *time of flight*. Beams are usually directed so that the widest part of the environment can be scanned with a reasonable angular precision. The high accuracy of theirs range measurements allows to retrieve the objects precise geometry. However, their relatively high cost and fragility slow down their deployment in many applications.

Radars

They detect the reflection of radiated electromagnetic energy (much higher frequencies than ladars) and measure ranges using either pulses time of flight, or frequency modulations between sent and received signals (Frequency Modulated Continuous Wave (FMCW) radars). The angular resolution of a radar is usually much lower than with ladars. As a result, they are usually unable to provide precise geometrical information. However, no other sensor can reach the range measure accuracy of a radar at long ranges and under adverse atmospheric conditions.

Figure 2.11: The ladar used in the experimental setup of chapter 6.

Sonars

They share the same basic *time of flight* principle but emit sound instead of electromagnetic waves. They typically operate around 45KHz. Sonars are usually cheaper than other types of range sensors but have a very limited angular resolution and are usually limited to a range of 10m.

Stereo-vision based sensors

They mimic the human visual system by making use of several cameras to infer depth information. This is done through the correlation of points of interests, patches or features in the images captured by the different cameras. The depth information is very dependent on pictures resolution, cameras calibration and light conditions and usually not very precise at long ranges. Another issue is the computational power that is needed to correlate in real time two or more images.

Vision sensors

While not being able to directly measure ranges, cameras are widely used sensors and the art of processing the images - computer vision - is a very dynamic research area. Data collected by cameras is complementary to the data collected by range sensors. Range sensors are blind to colours (except stereo-vision based sensors), textures and have much lower vertical and horizontal resolutions. Cameras are then very useful for classification. The most common cameras are sensitive to the visible spectrum and are as such limited to decent light conditions. Cameras operating in the far infrared spectrum are also widely used in robotics. They tend to be more expensive but offer interesting features like hot spot detection and allow night operations.

Figure 2.12: The camera used in the experimental setup of chapter 6.

2.3.2 Sensor modelling

Formalisation

As stated in the previous section, sensors collect data that are both *incomplete* (objects states are only partially measurable) and *imperfect* (measures are noisy). In order to make the most of the information sent by sensors, it is crucial to model the relation between objects true states and the corresponding observations made by sensors. Assuming that no uncertainties are involved, this relation can take the following form at time k:

$$Z_k = h_k(x_k)$$

, where

- x_k is the true object state,
- $Z_k = \left[z_k^1, z_k^2, ..., z_k^n\right]$ is the set of observation collected by the sensor from this object (there can be more than one measurement for one object),
- h_k is a nonlinear function from the state space to the observation space.

Uncertainties are usually modelled through an additive realisation of a random vector:

$$Z_k = h_k(x_k) + \left[w_k^1, w_k^2, ..., w_k^n\right]$$

, where $(w_k^i)_{1 \leq i \leq n}$ are n realisations of a random vector W_k. In a probabilistic framework, this relation is usually referred to as the following probability density $p(Z_k|x_k)$ usually called *measurements perceptual model*.

Range sensors modelling

For range sensors, a single measurement is expressed as:

$$z_k^i = \begin{bmatrix} r_k^i \\ \theta_k^i \end{bmatrix}$$

, where r_k^i is a range and θ_k^i is a bearing measurement. This measurement is usually directly modelled from the specific point $\widehat{z_k^i} = (\widehat{r_k^i}, \widehat{\theta_k^i})$ of the object that was measured by the sensor. A bivariate Normal distribution is then commonly used as a density for the noise vector W_k (Blackman & Popoli, 1999; Brown & Hwang, 1997).

$$W_k \sim N(\mu = \begin{bmatrix} \widehat{r_k^i} \\ \widehat{\theta_k^i} \end{bmatrix}, \Sigma = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_\theta^2 \end{bmatrix})$$

This model can then be transformed into the global reference frame using the following equations:

$$\begin{aligned} x_k^i &= x_{\text{platform}} + r_k^i \cos(\theta_k^i + \theta_{\text{platform}}) \\ y_k^i &= y_{\text{platform}} + r_k^i \sin(\theta_k^i + \theta_{\text{platform}}) \end{aligned} \tag{2.1}$$

When transformed through the above nonlinear equations, the bivariate normal distribution $p(z_k^i|\widehat{z_k^i})$ becomes the density function $p((x_k^i, y_k^i)|(\widehat{x_k^i, y_k^i}))$ depicted in figure 2.13. This density describes the likelihood that the real point that originated the measurement be in $(\widehat{x_k^i, y_k^i})$ given that the sensor made an observation in (x_k^i, y_k^i).

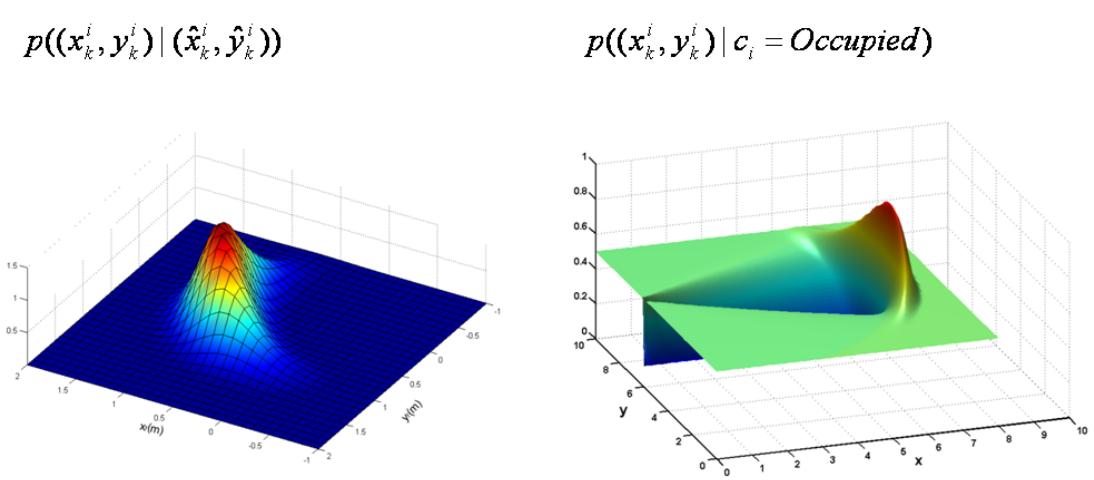


Figure 2.13: On the left: uncertainty model $p((x_k^i, y_k^i)|(\widehat{x_k^i, y_k^i}))$ in the global reference frame. On the right: the uncertainty model function of the positive occupancy of a point c_i in the global reference frame.

However, the sensor model sometimes appears in a slightly different form usually referred to as the *sensor occupancy model*. This density function represent the likelihood that a point c_i in the global reference frame is occupied having made a sensor observation (x_k^i, y_k^i). This former density contains additional information about the characteristics of the sensor regarding occlusions. The basic idea being that if a measurement is made in (x_k^i, y_k^i), the space between this point and the sensor is likely to be free while the area around that point is likely to be occupied. The sensor occupancy model that will be used in chapter 5 is shown

in figure 2.13. The computation of such model is out of the scope of this dissertation but more details can be found in (Elfes, 1989b; Papoulis & Pillai, 2002).

2.4 Environment representation

2.4.1 Foundations

Perceptual information continuously coming from sensors should be ultimately used to build an accurate environment representation. By considering that all objects can be represented as a vector in a *state space*, this environment representation should contain all the objects states current estimates produced by the perception algorithms.

But it should also contain information about the uncertainties related to these estimates. Note that because sensors will gradually acquire more and more measurements about the same objects in the scene, uncertainties are expected to decrease progressively. These observations lead to the following possible definition:

> *An environment representation is a knowledge base that contains the current estimates of some of the environment objects true states and the uncertainties related to them.*

An environment representation is indeed only supposed to contain estimates of some of the environment objects true states. First, no sensor is capable of observing all the state parameters of an object. Second, depending on the application, some of the observable state parameters are useless and as a result are not represented.

As mentioned before, each object state space contains in principle an infinite number of dimensions. It is impossible to represent entirely the full state of an object with a finite number of parameters. As a result, and because for a specific application it is not relevant to represent the full objects states, a much smaller set of objects features will be chosen and a knowledge representation will be built accordingly. For example a perception system intended to feed a collision avoidance device is likely to be only storing information about the position, speed and dimensions of the environment objects. However, even with reduced state spaces, optimal representation of the estimates are usually difficult to implement. This is mainly due to the fact that most objects parameters and uncertainties are continuous information which is usually difficult to store directly.

As a result, further simplifications are usually needed to build usable representations. Over the years, several options have been proposed that are all relevant trade-offs between these four criteria:

- Precision of the representation
- Precision of the uncertainty modelling
- Computation and storage requirements
- Capacity to represent the knowledge related to all the perception tasks.

A brief description of the most widely used environment representations is given in the next sections.

2.4.2 Common environment representations

Features based representation

In features based representations, objects states are drastically reduced to a small number of continuous or discrete parameters (e.g. cartesian position, speed, width and length of each object) that are hopefully representative enough for the considered application. The environment representation is then made of the list of these parameters estimates for each object and is as such very efficient in terms of storage requirements. Uncertainties on state parameters estimates are usually modelled using parametric probability density functions such as Normal distributions allowing memory efficient storage.

The main drawback of features based representations is arguably the fact that unstructured environment can not usually be precisely represented by low dimension state vectors. Features based representations are, as a result, usually used for well structured environments where objects can typically be correctly described by some geometrical primitives.

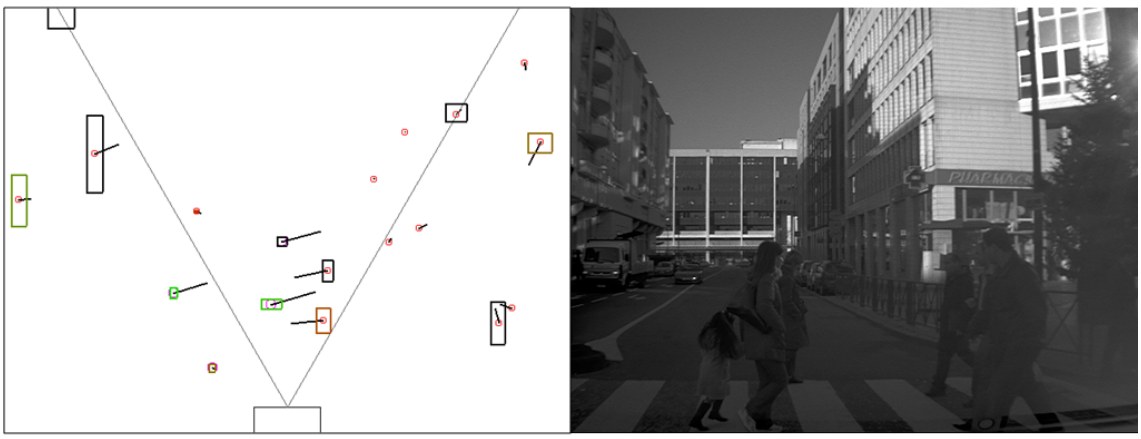


Figure 2.14: Example of a features based representation where the position, velocity vector and dimension of the outlining box is estimated for each object in the environment.

Grid based representation

In grid based representations, the environment is not directly described as a set of objects as for the features based representation. The environment is instead subdivided into an array or grid of rectangular cells and a probability value is stored for each of these cells. This value measures historically (Elfes, 1989a) the probability for that cell to be occupied but can in principle represents any other probability. The main advantage of this representation is to allow precise modelling of uncertainties for the state parameter that is represented. However, the memory requirements are usually high to reach sufficient resolutions for most applications. Some methods known as *quadtrees* (Kraetzschmar *et al.*, 2004) or *multiresolution grid map* (Montemerlo & Thrun, 2004) have been proposed over the years to address this problem and to allow lower memory usage.

Figure 2.15: Example of an occupancy map representation. White cells are free and black ones occupied.

Direct representation

When precise measurements are available, it is possible to represent the environment as a registered list of raw scans. Of course, the memory requirement for storing such a map grows very quickly. However, because scans usually overlap, it is possible to regularly simplify the point cloud using decimation. The representation of object shapes and positions can be very precise but uncertainties are usually not modelled at all. Besides, this representation only stores information about the object features that the sensor directly measures which limits the use of this type of representation to low interpretation level (i.e position and geometrical features). More details on this type of representation can be found in (Gutmann & Schlegel, 1996) and (Chen & Medioni, 1991).

Gaussian based representation

Gaussian based representations have been proposed as a kind of intermediary solutions between grid based and direct representations. The idea is to represent uncertainties about object positions as sums of Gaussians. In (Bailey, 2002), a scaled Gaussian is centered on each sensor measurement to model sensors uncertainties. Even if the resulting density has not a clear mathematical signification, efficient methods based on gradient descent can be used for scans registration. However, the problem of merging multiple registered scans is not addressed. In (Biber, 2003), a similar Gaussian based approach is proposed, but the number of Gaussians present in the sum is contained. Like occupancy grids, a subdivision of the environment is established. In cells where at least three sensor measurements were made, a Gaussian is initially created. Then, each Gaussian in the sum is iteratively updated using the new measurements that fall into the corresponding cell. The known gradient of Normal densities can again be used for scan registration while the memory growth over time is contained. This interesting environment representation is limited to geometrical map representation and cannot directly store detection, classification or tracking results. Besides, it is important to note that free spaces are not directly represented (unlike occupancy grids).

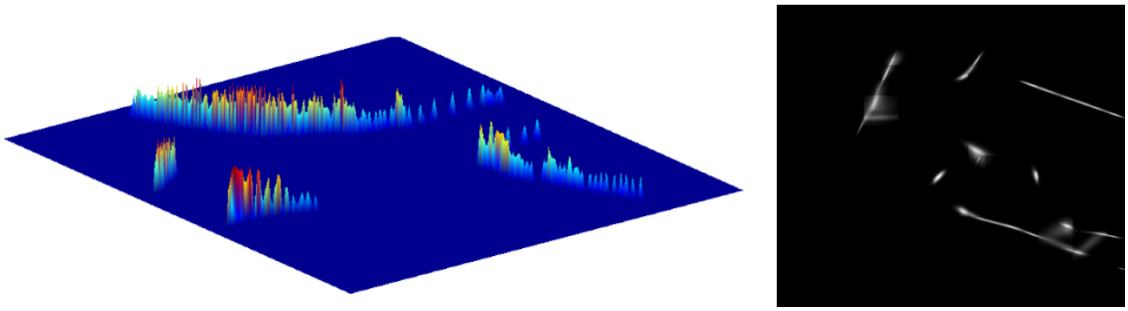


Figure 2.16: Example of environment representation using variable number of Gaussians (Bailey, 2002) (on the left) and a fixed number of Gaussians (Biber, 2003) (on the right).

Hybrid representation

Because no paradigm is completely satisfactory, it is possible to use at the same time different map representations. The objective is to benefit from the advantages of each of them. In (Wang & Thorpe, 2004) an hybrid environment representation called *hierarchical object based representation* is proposed. A direct representation is used to register the successive scans, an occupancy grid based map is used to store the geometrical features of the objects and a feature representation is employed to store objects position and velocity estimates and to construct large scale representations.

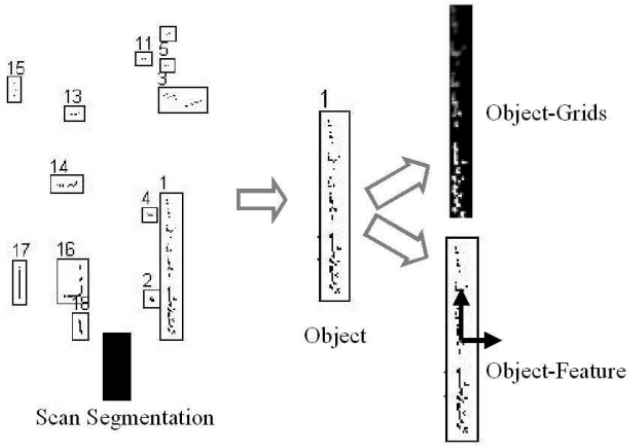


Figure 2.17: Example of an environment represented through the *hierarchical object based representation* proposed in (Wang & Thorpe, 2004).

2.5 Multiple Object Detection, Tracking and Classification

In this section a brief summary of the principles and technics used to detect, track and classify static and moving objects is given. Over the last decades a massive amount of different methods based on all types of sensors have been proposed. As a result this review cannot pretend to be exhaustive. However, most of these proposed algorithms are in fact variants of a small number of influential approaches. More than enumerating all the existing technics, this section is thus intended to present the few main paradigms under which can be grouped the vast majority of past and current research works in multiple objects detection, tracking and classification.

It is important to note that the detection, tracking and classification of all the objects - static or moving - in the environment is considered. While detection and tracking applied to moving objects is usually referred to as *DATMO* algorithms for *Detection And Tracking of Moving Objects* (Wang *et al.*, 2007; Vu & Aycard, 2009; Benenson, 2008), the following abbreviation ***DETAC*** for any type of object *DEtection, Tracking And Classification* is preferred here and will be used in this dissertation.

2.5.1 Tracking

Even if tracking is usually not the first task to be solved in the data flow, this task is the best formulated and the one that received historically the best attention. In their early days, tracking technics were indeed employed for defense applications in very specific environments and detecting or classifying air fighters or cruisers in unobstructed environments was less problematic than estimating their real trajectories. As a consequence, tracking is usually solved independently from the other tasks making the assumption that objects are correctly detected beforehand. In the vast majority of existing *DETAC* systems, the perceptual tasks are solved sequentially in that order: detection, tracking and then classification. As will be seen in chapter 4, this lead to sub-optimal algorithms and solutions have recently been proposed.

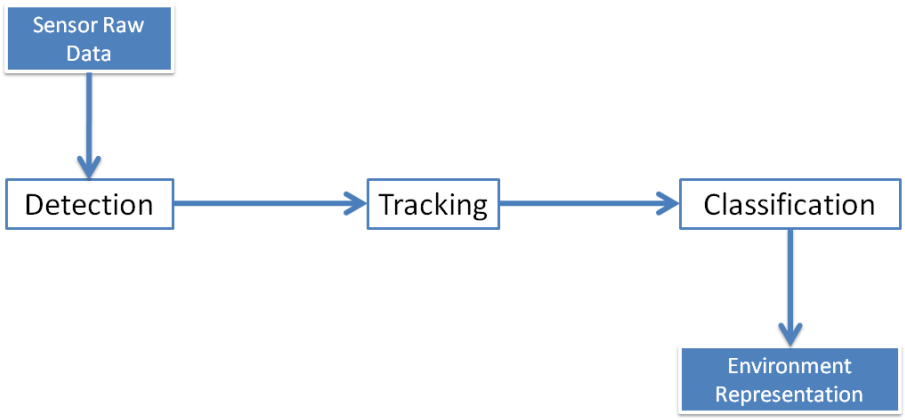


Figure 2.18: Data flow used in most objects detection, tracking and classification systems (*DETAC*)

As mentioned at the beginning of this chapter, the tracking problem is the combination of two important and different sub-problems: the filtering problem and the data association problem.

Filtering

A filtering algorithm aims at (iteratively) estimating the real state of an object based on successive sensor measurements related to that object. This problem can be formulated on various forms but the most commonly used in the autonomous perception field is certainly the Bayesian formulation of this problem following the influential work of R.E. Kalman (Kalman, 1960).

The object state vector is usually described as a random variable noted $X_k \in R^n$ and its corresponding measurement $z_k \in R^m$. The problem of estimating X_k from all the measurements collected up to time k is then regarded as estimating the density $p(X_k|Z_{0:k})$, where $Z_{0:k} = \{z_0, ..., z_k\}$. This estimation can be formally written as follows:

$$p(X_k|Z_{0:k}) = \underbrace{\frac{p(z_k|X_k, Z_{0:k-1})}{p(z_k|Z_{0:k-1})}}_{\text{Correction Term}} \underbrace{\int_{X_{k-1}} p(X_k|X_{k-1}, Z_{0:k-1}) p(X_{k-1}|Z_{0:k-1}) dX_{k-1}}_{\text{Prediction Term}} \tag{2.2}$$

Unfortunately, there exist no closed form solution for this equation without using additional assumptions on the state dynamics $p(X_k|X_{k-1}, Z_{0:k-1})$, sensor model $p(z_k|X_k, Z_{0:k-1})$ and on the uncertainty representation. Depending on these assumptions, a great amount of famous solutions have been proposed that can be grouped into two groups: single model and multiple models approaches.

Single model approaches

When the object state X_k verifies the Markov property, the general form of the state dynamics can be simplified as follows:

$$p(X_k|X_{k-1}, Z_{0:k-1}) = p(X_k|X_{k-1}) \tag{2.3}$$

This means that X_k only depends on X_{k-1} and on some independent noise random variable. This simplification usually holds when a single dynamic model (constant velocity model, constant acceleration model, bicycle model, etc...) is employed, hence the name given to these approaches. Precisely describing the various technics that are used to solve the filtering problem in that case is both out of the scope of this thesis and already widely discussed in the literature.

To name just a few of them, the Kalman filter addresses the filtering problem with the additional assumption that the state dynamics are linear and the independent noise Gaussian. The extended Kalman and the unscented Kalman filters are both sub-optimal variants that handle nonlinear dynamics. The particle filter, a sampling (Monte-Carlo) based approach has been given a lot of attention in the last decade. This former approach allows to consider both nonlinear dynamics and non Gaussian noises (Herman, 2002; Arulampalam *et al.*, 2001).

Multiple models approaches

Depending on the application, single model dynamics might be insufficient to model the behavior of an object. This is especially true when the object is prone to quick behavior changes (walking, stopping, turning, running, etc...). In that case, an additional discrete random variable $r_k \in \mathcal{R}$ can be used to drive X_k using multiple models (one for each simple behavior). The simplification of the equation 2.3 is not valid anymore and has now to be derived as follows:

$$p(X_k|X_{k-1}, Z_{0:k-1}) = \sum_{r^i \in \mathcal{R}} p(X_k|X_{k-1}, r_k = r^i) \quad p(r_k = r^i|Z_{0:k-1}) \tag{2.4}$$

As a consequence, even if the noises are Gaussian and every single model linear, the filtering problem stays significantly more difficult than in the corresponding single model case. Indeed, the marginalisation over the motion mode r_k leads overtime to highly multi-modal densities (exponentially growing mixture of Gaussians in the Gaussian case). If particle filters can in principle deal with that problem, ensuring a correct description of such multi-modal densities with particles is a difficult task. Hence, more appropriate methods have been developed. Most common approaches use a fixed number of Gaussian to approximate densities. Depending on when and how these Gaussians are merged to contain the exponential growth, these technics are called *Generalised Pseudo-Bayesian approaches (GPB)* (Tugnait, 1981) or *Interactive Multiple Model (IMM)* (Blom & Bar-Shalom, 1988) approaches. More details about these technics can be found in (Andrieu *et al.*, 2003) and (Wang, 2004).

Data association

All the filtering methods presented above assume that the correct observation is retrieved at each time step for every tracked object. In practice, associating incoming measurements with the appropriate tracked object is a complicated task. It is possible to categorize the existing approaches depending on the nature of the result they produce in terms of uncertainty management. Indeed, associating an observation with an already tracked object cannot be done with certainty and dealing with this uncertainty can be done in several ways.

Full Posterior Densities approaches

Optimal solutions should of course produce a full posterior probability density over all the possible associations between an object and the current sensor observations. Unfortunately, this is intractable in most situations as it requires in principle to marginalize on all the past possible associations (usually a huge number of possibilities). Indeed, the correct association between objects and measurements depends on both current and past measurements and not just on the previously computed associations (to put it differently, the considered random process is not Markovian).

A first simplification known as *Joint Probabilistic Data Association Filter (JPDA)* (Bar-Shalom & Fortman, 1987; Schulz *et al.*, 2001) consists in considering only the possible

associations with the current measurements and computing as a result a *single scan* sub-optimal posterior density about the possible associations. In the filtering process, the state density $p(X_k|Z_{0:k})$ is then obtained by summing (marginalizing) over the weighted possible associations with current measurements.

A more recent simplification known as *Markov Chain Monte-Carlo Data Association (MCMCDA)* (Zhao *et al.*, 2008; Song & Nevatia, 2005; Oh *et al.*, 2004) aims at estimating the full posterior density through sampling. The general idea of Monte-Carlo approaches is to compute estimates of probability densities under the form of a set of weighted samples. Under specific assumptions, it is indeed possible to draw samples of a density without knowing it explicitly. In other words, the density of interest is estimated through the estimation of the probability of some well chosen association hypothesis (samples) instead of a greedy and impossible enumeration of all the possible hypothesis.

Maximum A Posteriori (MAP) approaches

Instead of trying to estimate the full posterior association density, it is also possible to find only the most probable association hypothesis (the hypothesis that maximises the above full posterior density). In that case, the computation of the state density $p(X_k|Z_{0:k})$ is simpler as only one association hypothesis is considered. However, it still requires that all possible associations over time be considered and is thus subject to growing complexity. This approach was introduced by (Reid, 1978) under the name of *Multiple Hypothesis Tracking (MHT)* and have been adapted ever since to control its growing complexity (Blackman, 2004).

Maximum Likelihood (ML) approaches

Finally, a widely used set of methods assume that correct association can be obtained without considering any past measurements nor previous knowledge about possible associations. These methods compute the association hypothesis that best fits the current measurement. In a Bayesian framework, this kind of approaches are said to maximise the measurements likelihood. The *Nearest Neighbor* (Blackman & Popoli, 1999) technics is a typical *ML* approach that consists in finding the association that minimizes a distance between objects and associated measurements. If this method is obviously less robust than those presented above, its computational requirements are very low and its efficiency acceptable in many situations. As result, this is still a widely used technic for real time perception systems.

2.5.2 Detection

As seen above, the tracking algorithm needs to be feeded regularly with state variables measurements. Unfortunately, sensors do not provide this information directly. Instead a great number of unordered raw points is collected that need to be grouped before being associated with tracked objects. Categorizing detection technics using the statistical nature of the algorithm is not possible here as most existing methods do not handle uncertainties. It is indeed important to notice that *detection* is not a problem that is as well mathematically formalised as *tracking*. However, detection approaches can be classified in two groups : *geometry based methods* and *behavioral based methods*.

Geometry based methods

A widely used approach is to use heuristics about the spatial repartition of the raw data to infer object detection. Methods that fall in this class are of two types. *Model free detection technics* (Fayad & Cherfaoui, 2007; Schulz *et al.*, 2001; Gate & Nashashibi, 2009) use simple distance criteria between raw data to segment the scan in what is supposed to be objects. On the contrary, *Model based detection technics* make use of a predefined model about the particular objects that should be detected and select the raw data sets that best fit with this model. While being more robust than the *model free methods* in most cases, they tend to be limited to the detection of highly structured objects that can easily fit with a specific model. An interesting example of a model based pedestrian detection through a kernel function based model is given in (Gidel *et al.*, 2008).

Behavioral based methods

A good manner to refine detection is to look at raw data evolution over time. Unfortunately, dynamic information (such are velocities or accelerations) are not measured by sensors (except for Radars) and has to be inferred. Tracking algorithms are specifically designed to solve these tasks but initiating trackers on each raw data is not feasible in practice as the data association would be impossible to solve due to the fact that none of the successive raw points are related to the same real object points over time. This problem is arguably one of the most important cause of failure in *DETAC* systems and has paradoxically received a limited attention in the last decade compared to tracking issues.

Without being able to completely discriminate all the objects in the scene, methods have been proposed to classify raw data as coming from static or moving objects using simple heuristics. Indeed, given a correct localization of the sensing platform, it is possible to detect that a group of raw data is currently located in a place that was unoccupied in the previous scans. Moving object detectors of this type are employed in (Wang *et al.*, 2007) or (Wolf & Sukhatme, 2005) for example but are recognised in (Wang *et al.*, 2007) to be prone to limited performances when objects are moving slowly. Besides, such detector do not solve the problem of discriminating moving objects from one another. A simple geometrical criteria is then usually used to do so and as such inherits from the limitation of the approaches presented above.

However, insightful work has recently been done in (Vu & Aycard, 2009) that built on (Petrovskaya & Thrun, 2008) to propose an interesting and viable solution to this problem. Tracking and detection are performed simultaneously using a predefined shape model for every considered type of objects. A *MAP* estimate of this joint inference problem is computed efficiently using a *MCMC* approach. If this approach is limited to objects that can effectively fit with basic primitives and required the use of a pre-detection routine intended to detect parts of objects, it certainly is a novel and promising direction. In chapter 5, a model free approach to that problem is proposed that shows similar properties.

2.5.3 Classification

Most applications requires that the detected and tracked object be classified. Classification strategies fall in two categories: *Heuristics based methods*, and *learning based methods*.

Heuristics based methods

Provided that some relevant object features can be estimated through detection or tracking (dimensions, velocity, etc...), it is possible to use experimental rules to classify objects. The main advantage of this approach is that it is both simple to implement and computationally efficient. But there are at least two main drawbacks. First, as these heuristics are based on high level features that are not directly measured by sensors, performances are highly dependent on the estimators (detection or tracking algorithms) that produce the employed features estimates. Second, it is in practice often difficult to find a set of rules that are specific enough to filter undesired objects and vague enough to adapt to all the different types of objects that exist within a class (big cars and small cars have both to be classified as "vehicles"). Examples of such rules can be found for pedestrian detection in (Fayad & Cherfaoui, 2007; Fayad *et al.*, 2008; Gate *et al.*, 2009) and for vehicle detection in (Petrovskaya & Thrun, 2009; Fayad & Cherfaoui, 2007).

Learning based methods

Due to the fundamental limitations of the heuristics based methods mentioned above, a different approach has been given a lot of attention over the last three decades in all computer science fields where classification or recognition problems are considered. Instead of finding explicit rules to classify entities, these approaches are based on the idea that these rules can be found automatically. In the autonomous perception domain, learning routines are very rarely applied on unprocessed raw data. Instead, relevant features are estimated that are then used by the learning algorithm to find the optimal classifier. Besides, the learning process is made off-line in a supervised way (labelled data) in most cases. While significantly more demanding in terms of implementation, these approaches have been proven to be far more efficient than heuristics in lots of applications. A huge number of such classification methods has been proposed in the recent literature. For example, implementations for ladar based pedestrian classification can be found in (Spinello *et al.*, 2008) and (Wender *et al.*, 2005).

2.6 Simultaneous Localization and Mapping

2.6.1 Principles

The localization and mapping problem are two perception tasks that are highly dependent on one another and are as such usually addressed as a joint inference problem of the following density:

$$p(X_k, M_k | U_{0:k}, Z_{0:k}) \tag{2.5}$$

, where $M_k = \{m_k^1, m_k^2, ..., m_k^n\}$ is random vector representing the map of the environment (as a list of so called *landmarks*), X_k is the global position and speed of the sensing vehicle at time k, $U_{0:k} = \{u_1, u_2, ..., u_k\}$ are the measurements coming from the proprioceptive sensors and $Z_{0:k} = \{z_0, z_1, ..., z_k\}$ are the measurements from the perception sensors. The *SLAM* problem can then be written as a recursive Bayesian inference problem:

$$\underbrace{p(X_k, M_k | U_{0:k}, Z_{0:k})}_{\text{Posterior}} \propto \underbrace{p(z_k | X_k, M_k)}_{\text{Measurement likelihood}} \underbrace{\int_{dX_{k-1}} \int_{dM_{k-1}} p(X_k, M_k | X_{k-1}, M_{k-1}, u_k)\ p(x_{k-1}, M_{k-1} | Z_{0:k-1}, U_{0:k-1})\, dM_{k-1} dX_{k-1}}_{\text{Prior}} \tag{2.6}$$

In this form, this inference problem is very difficult as it requires a marginalization (weighted summation) over all the possible maps M_{k-1} whose corresponding space is of high dimensionality. Besides, in practice moving objects are often too unpredictable to help localization. These two observations lead to the static world assumption used by the vast majority of mapping algorithms. Making the additional assumption that the sensing platform moves independently from the map (which is not always true in practice), a simplified form is obtained.

$$p(X_k, M | U_{0:k}, Z_{0:k}) \propto \underbrace{p(z_k | X_k, M)}_{\text{Measurement perceptual model}} \int_{dX_{k-1}} \underbrace{p(X_k | X_{k-1}, u_k)}_{\text{Sensing platform motion model}} p(X_{k-1}, M | Z_{0:k-1}, U_{0:k-1}) dX_{k-1} \tag{2.7}$$

Similarly to tracking technics, the existing methods can be categorized on the nature of the computed density estimate. A brief overview of the most common approaches is given in the following section. For a more detailed review, the interested reader can refer to (Thrun, 2002).

2.6.2 Common Technics

Full posterior density approaches

Depending on the assumptions made on the nature of the motion model, the perceptual model and the posterior density that has to be estimated, two main paradigms are used.

When models are assumed to be linear and noises and successive posterior densities Gaussians, classical Kalman equations can be employed. This approach known as *Kalman Filter SLAM* makes use of features based representations of the static map where static and easily distinguishable landmarks have to be selected and sequentially associated with new incoming measurements. This approach is very much similar to Kalman filter based tracking approach and suffers from the same limitations: the Gaussian and linear assumptions can be too restrictive (sensing platform motion model are usually nonlinear) and the association problem has to be handled separately. These two limitations lead to practical implementation difficulties as using a growing set of landmarks brings association ambiguities and using too few of them does not produce very rich maps and can lead to localization ambiguities incompatible with the unimodal Gaussian assumption.

Quite logically, the same Kalman filter extensions used in the tracking problem are employed here to accommodate some nonlinearities and in particular a well known particle filter (Monte-Carlo) based approach known as *FastSLAM* allows nonlinearities and free form densities to be handled efficiently. In its pure form, the particle filter would consist in sampling the joint density $p(X_{0:k}, M|U_{0:k}, Z_{0:k})$. The considered sample space would contain many dimensions (depending on the number of *landmarks*) and would lead to limited computational efficiency. *FastSLAM* earned its name by using an exact factorization of the joint probability:

$$p(X_{0:k}, M|U_{0:k}, Z_{0:k}) = \underbrace{p(M|X_{0:k}, Z_{0:k})}_{\text{Map estimation with known poses}} \underbrace{p(X_{0:k}|U_{0:k}, Z_{0:k})}_{\text{Poses estimation}} \tag{2.8}$$

, and by using the Rao-Blackwell theorem ensuring that in this product if one of the two terms can be computed analytically, sampling the remaining term suffices to compute samples of the joint density. In other words, the *SLAM* problem can be reduced to the combination of a sampling problem in a reduced space and an analytical mapping with known poses problem. This approach is proved to be significantly faster than Kalman filter based approaches allowing much more landmarks to be added to the map and hence better environment representation.

Maximum Likelihood approaches

Instead of trying to compute an estimate of the full joint density $p(X_{0:k}, M|U_{0:k}, Z_{0:k})$, algorithms have been proposed to compute at each time step the most likely map and robot pose. In particular, the *expectation-maximisation (EM)* family of algorithms have received lots of attention for maximum likelihood problems with what is called *latent variables*. Indeed, finding the most likely map $\widehat{M}$ can be written as the following maximisation problem:

$$\widehat{M} = \arg\max_{M} p(Z_{0:k}|M) \tag{2.9}$$

However, maximising the above likelihood is usually intractable as it hides a missing (or *latent*) variable: the robot pose. Computing the above problem would indeed require a marginalization over all the possible robot trajectories. *EM* algorithms consists in computing

a series of map estimates $\widehat{M}^{[0]}, \widehat{M}^{[1]}, ..., \widehat{M}^{[i]}$ that converge to the most likely map $\widehat{M}$. This is done through the iterative computation of two steps.

1. Expectation step: the expected value of the log likelihood function is computed with respect to the current estimation of the robot pose posterior given the current estimation of the map:
$$Q(M|M^{[i]}) = E_{X_k|M^{[i]},Z_{0:k}}\left[\log p(Z_{0:k}, X_k|M)\right] \tag{2.10}$$

2. Maximisation step: the new map estimate is computed from the maximisation of the function computed in the previous step.

$$M^{[i+1]} = \arg\max_{M} Q(M|M^{[i]}) \tag{2.11}$$

Various methods are used to solve the maximisation step and to compute successively estimations of the robot pose posterior given the current estimation of the map. The main advantage of the *EM* approach over the full posterior approaches described above is that the association problem is implicitly handled. Using the robot pose posterior density in the E-step allows to maintain different hypothesis about where the robot might be, hence different association hypothesis. While bringing an elegant solution to the data association problem, *EM* algorithms are not well suited for online implementation (because of their iterative nature) and simplifications are commonly used.

A very popular and much simpler *ML* approaches called *incremental maximum likelihood* methods can be seen as an *EM* algorithms with no *Expectation steps*. At each time step k, an estimate of the map $\widehat{M}_k$ and of the robot pose $\widehat{X}_k$ using previously made estimates $\widehat{M}_{k-1}$ and of the robot pose $\widehat{X}_{k-1}$ is computed.

$$\left\langle \widehat{M}_k, \widehat{X}_k \right\rangle = \arg\max_{M_k, X_k} p(Z_{0:k}|M_k, X_k)\ p(M_k, X_k|\widehat{M}_{k-1}, \widehat{X}_{k-1}) \tag{2.12}$$

Because uncertainties related to the robot pose are lost at each new computation, this method is less robust than *EM* approaches. It is consequently difficult to map large cyclic environments using this method as errors in robot pose is likely to grow over time and to prevent the system from being able to "close the loop". However, this approach is fast and simple and has been proven to work well in practice.

2.7 SLAM with DATMO

As shown in the last two sections, significant efforts have been carried out to propose satisfactory solutions for every perceptual task. Because of their direct and explicit interdependencies, localization and mapping have historically been addressed as a joint inference problem assuming that the world is static in most situations. At the same time, with less explicit interdependencies, the three remaining perceptual tasks: detection, tracking and classification have been addressed separately as independent problems by a slightly different scientific community.

Recently, significant efforts have been made to fill the gap that still exists between these tasks. The *SLAM* community is slowly heading to dynamic environment mapping problems while a part of the *MOT* community is looking at mobile object mapping and environment based localization routines. *SLAM* and *DATMO* have been recognized as mutually beneficial and pioneering approaches have been proposed to compute these tasks together.

All of them propose original methods to compute *SLAM* with *DATMO* (classification is not addressed directly in these works) based on the idea that a robust localization of the sensing platform can be used to efficiently detect moving objects in the scene while this latter knowledge allows in return better localization and static objects mapping (moving objects can be filtered out before SLAM computation). A brief overview of these works is given in the next paragraphs.

Interesting systems have been first proposed to deal with indoor environments. In (Prassler *et al.*, 1999), a dead-reckoning localization (localization without the help of external sensors) is performed to construct over time a grid based map of the stationary objects. At the same time a simple heuristic is used to detect mobile objects and a analytical tracking scheme is initiated on each of them. While being quite simple, this approach was proven to be efficient in indoor environments with slow moving sensing platforms.

In (Hahnel *et al.*, 2003) and (Montesano *et al.*, 2005), two similar systems are proposed with however a more probabilistic approach. Incremental maximum likelihood methods are used for localization and mapping while extended Kalman filter (Montesano *et al.*, 2005) or particle filters (Hahnel *et al.*, 2003) are used to track moving objects. Perceptual knowledge is stored in an occupancy map for static objects and in a features based map for mobile objects. Similarly to (Prassler *et al.*, 1999), mobile object detection is performed by detecting new sensor observations that fall into spaces previously unoccupied (free space violation). However, this simple heuristic is integrated in a Bayesian framework. Indeed, every new sensor observation is given a probability to be related to a mobile object. During SLAM computation, the use of every sensor observation is weighted by this probability, making the whole process less dependent on mobile objects detection failures.

At the same time, (Wang, 2004) proposed a comparable system whose main originality lies in its ability to scale to large outdoor. This is achieved by using a relevant mix of a direct, grid and features based environment representation called *hierarchical object based representation*, a direct *ML* method for SLAM (scan matching through an *Iterative Closest Point (ICP)*

based algorithm) and a *IMM* based mobile objects tracking. Although the detection of moving object is also based on free space violation, this classification is not embedded in a Bayesian framework and is as such a sensitive part of the system.

More recently another insightful SLAM with DATMO approach has been proposed by (Vu & Aycard, 2009). While being very similar to Wang, except that SLAM is computed using an incremental *ML* method, the proposed method makes use of a sampling based approach to solve simultaneously the detection and tracking of moving objects. A computer efficient *MAP* estimate of the detection and tracking joint density is computed by using a *Markov chain Monte-Carlo (MCMC)* sampler as already mentioned in paragraph 2.5.2 of this dissertation.

All the systems presented in this section compute SLAM and DATMO at the same time but with still noticeable boundaries between the five perceptual tasks. Indeed, these systems can arguably and mostly be seen as a juxtaposition of a SLAM algorithm and a DATMO algorithm that share limited information. In other words, there are still important boundaries between localization and mapping on the first hand and detection, tracking and classification on the other hand.

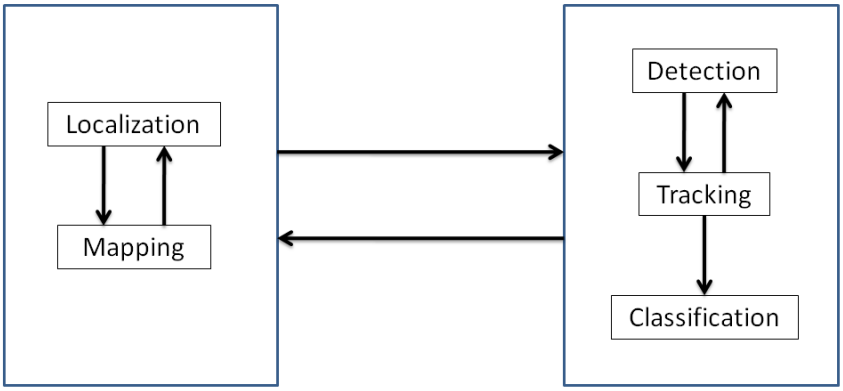


Figure 2.19: Schematic view of current *SLAM* with *DATMO* algorithms.

The main implication is that all these systems require a clear distinction between sensor observations that belong to static objects and those belonging to moving objects. In practice such a distinction might be difficult for slow moving object like pedestrians as described in (Wang *et al.*, 2007). Besides, only a small number of possible interactions between the tasks is exploited.

2.8 Conclusion

In this chapter, a brief but hopefully clear overview of existing autonomous perception algorithms is given. As seen in the last section, significant efforts have been recently carried out to put efficient SLAM algorithms and DATMO algorithms together in the same system. Interesting approaches have been proposed but the interactions between the five perceptual tasks are still very limited. Indeed, in all SLAM with DATMO systems, SLAM provides to

DATMO a refined localization and a rough classification of sensors measurements as moving or static. Symmetrically, DATMO provides to SLAM a list of currently tracked objects that should not be incorporated in its next computation.

Making use of such an interaction is undoubtedly a good way to enhance globally the capabilities of perception systems. However, we believe that this effort should be extended to design a perception system where localization, mapping, detection, tracking and classification are computed within the same mathematical framework allowing every possible interaction between the tasks to be elegantly exploited.

To objectively quantify the benefit of such an approach (presented in Chapter 5 and 6 of this dissertation), it is critical first to analyze what state-of-the-art perception algorithms such as those described in this section can do and most importantly what they cannot achieve in practice. Using the pedestrian perception problem in urban environment as a test-bed, we investigate in the following chapter what can be achieved with common perception approaches.

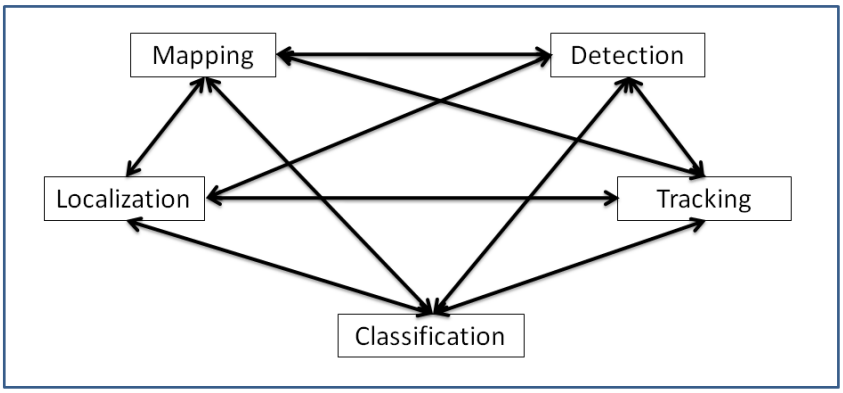


Figure 2.20: Schematic view of what can be achieved with the approach proposed in Chapter 5.

Résumé en français du chapitre 3

Ce chapitre présente de manière objective un système de perception complet représentatif, nous l'espérons, de ce qu'il est possible de construire avec les algorithmes déjà proposés dans la littérature. Ce système basé sur la contribution d'un capteur LIDAR et d'une caméra reste cependant innovant dans la façon dont les incertitudes sont traitées et dans sa capacité à "percevoir" les piétons et les groupes de piétons. Nous présentons en détail les performances de ce système qui serviront de "références" dans la suite de ce mémoire.

Chapter 3

Pedestrian Perception - Fast System

Contents

3.1 Introduction

We present in this chapter a system that addresses the problem of pedestrians perception in urban environments. As mentioned earlier, the work presented in this chapter is expected to meet two objectives:

1. Design a fast pedestrian perception system from state-of-the-art approaches.
2. Produce quantitative results on the performances of such a system to allow an objective analysis about what such systems can achieve in terms of reliability.

It is worth noting that the algorithm presented in this chapter is now an important contribution to a research project called *LOVe*[1] ("**L**ogiciels d'**O**bservation des **V**ulnerabl**E**s") sponsored by the French government and accompanied by two companies: *RENAULT* (car manufacturer) and *VALEO* (automotive supplier).

[1]The LOVe project is under the coordination of Professor L. Trassoudaine, LASMEA, 63177 Aubiere, France

3.2 Existing systems and related work

3.2.1 Pedestrian perception systems already commercialized

In the last five years, some pedestrian perception systems have started to be commercialized by the automotive industry to serve pre-crash purposes. These systems, equipping usually high-end cars, have some strong limitations that are mainly due to the nature of the sensors they are based on.

Night vision systems

The vast majority of pedestrian perception systems that have appeared on vehicles in the recent years are based on infrared cameras. By being sensitive to light in the infrared or ultra-violet spectrum, these sensors are able to "see" objects that are normally not visible at night to human eyes. Some of these sensors are aided by an appropriate infrared illumination of the scene. Because objects that are particularly hot naturally emit infrared radiations, these sensors present interesting capabilities. Objects detection and classification are consequently made much easier (as seen in figure 3.1) by using such sensing strategies. Unfortunately, all these advantages disappear during daylight where every kind of object in the scene is likely to emit infrared radiations. Besides, as explained in the previous chapter, it is not clear how a precise range information can be extracted from systems using a single monocular camera. Almost all automotive brand that produce high-end cars have developed their own night vision system based on that principle.

Figure 3.1: The night vision system of a famous german automotive company.

Daylight systems

There are a much lower number of commercialized pedestrian perception systems that work at the same time on daylight and on mobile platforms. One of them is designed by a company called *Mobileye* and will presumably be launched in 2010 with the contribution of the

automotive company *Volvo*. This system will apparently be based on the contribution of a radar and a monocular visible camera.

Communication based systems

It is interesting to mention an effort that is currently undertaken by the automotive company *Nissan* to address the problem of pedestrian safety in a original way. Instead of trying to detect pedestrians from *perceptual* sensors (cameras, ladars, radars, etc...), this firm is investigating the possibility to detect and locate pedestrians through their cell phones. This approach would have the big advantage to solve at the same time all the classic perceptual tasks by relying directly on signals transmitted by surrounding pedestrians. This approach is however limited to pedestrians lucky enough to be equipped with appropriate cell phones.

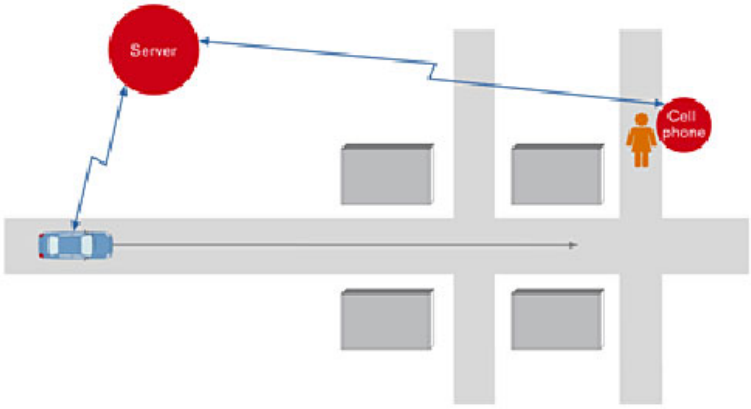


Figure 3.2: Schematic view of the pedestrian approach adopted by *Nissan*.

3.2.2 Pedestrian perception in French and European research projects

In the recent years a significant number of research projects implying both academic and industrial entities have been elaborated and have generated a great variety of work related to pedestrians perception with various sensor strategies. A non-exhaustive list is given below:

- PROTECTOR[2] (2000-2002) and SAVE-U[3] (2002-2005) led by Gravila and his team at Daimler AG.
- INTERSAFE[4] (ending in 2007) led by IBEO.
- WATCHOVER[5] (2005-2008) related to pedestrian safety through communication and vision based strategies.
- CAMELLIA (ended in 2005)

[2] www.gravila.net
[3] www.save-u.org
[4] www.prevent-ip.org
[5] www.watchover-eu.org

- LOVE[6](2006-2009)
- AKTIV[7] (2006-2010)

These research projects that involve(d) some of the major european automotive companies show the interest of both industrial and governmental authorities into pedestrian safety.

3.2.3 Pedestrian perception systems based on ladar in the literature

In the scientific literature, a significant number of complete pedestrian perception systems have been proposed over the last decade. In practice, because the accurate estimation of object range is a prerequisite to efficient collision avoidance systems, these solutions are very often based on ladar sensors. As ladar will also be employed in the system described in this chapter, we give in this section a brief overview of some of the pedestrian detection systems based on ladar that proved to be usable in real time.

All these methods can of course be differentiated from the approaches chosen to solve sequentially the detection, tracking and classification problems. However, we believe that the classification method that is employed accounts for the most part of these systems respective performances. Indeed, the detection methods are usually very much similar to one another (usually based on some distance criteria) and perfect tracking is not directly of paramount importance for such applications. Indeed, missing the fact that two pedestrians sequentially detected are in fact the same real pedestrian is not crucial provided that dynamical features are still well estimated to maintain correct classification. In this overview, existing systems are consequently categorized from the classification strategy they propose.

Heuristics based pedestrian classification

In (Fuerstenberg *et al.*, 2002) and (Streller *et al.*, 2002) the current dimensions of the detected objects are used for classification. However, because objects can be occluded at certain point in time, these approaches have a limited efficiency in complex scenes. A natural turnaround is to incorporate such geometry based classification rules into a Bayesian framework.

In (Zhao *et al.*, 2006; Premebida & Nunes, 2006; Mendes *et al.*, 2004; Fayad & Cherfaoui, 2007) a Bayesian filter is used to iteratively update a posterior probability mass function over the possible objects classes. The dimensions of the currently detected objects are then incorporated in the filter as a likelihood. Consequently, if an object is temporarily occluded and its visible dimensions modified, the system can still ensure accurate objects classification.

When pedestrian legs are visible on ladar raw data (this depends mainly on the object range and on the angular resolution of the sensor), a classification can be made on this feature as shown in (Xavier *et al.*, 2005; Shao *et al.*, 2007; Cui *et al.*, 2007; Zhao & Shibasaki, 2005; Fuerstenberg *et al.*, 2003). All of the above approaches use a ladar to classify pedestrians

[6]http://love.univ-bpclermont.fr
[7]www.aktiv-online.org

but interesting shape based classification method can also be used with cameras as shown in (Bertozzi *et al.*, 2003).

Learning based pedestrian classification

As mentioned in chapter 2, over the years learning based classification methods have been proven to be more efficient than heuristics based technics in many applications. This is particularly true when sensors provide rich information content, usually complex to handle through heuristics. Consequently, learning based methods are much more common in computer vision than in ladar based processing. However, some interesting efforts have been made to apply learning based approaches to ladar classification.

In (Spinello *et al.*, 2008), a cascade of support vector machines is used to classify clustered raw data. In (Wender *et al.*, 2005), a trained neural network is used to classify objects from extracted features. In (Zivkovic & Krose, 2007) a trained leg detector is used for classification. Finally, in (Arras *et al.*, 2007) a boosting algorithm based on a set of geometrical features extracted from clusters is employed. Even if these approaches give satisfying results, their benefits over simpler classification schemes for ladar data classification is not well established.

Some multi-sensor strategies have also been recently proposed. In (Spinello *et al.*, 2008), the classification scheme mentioned above is combined with an *implicit shape model (ISM)* based classification method for refining object classification using the corresponding region of interest in the camera image.

3.3 Principles

3.3.1 Requirements

As mentioned in Chapter 1, a perception system intended to feed an onboard collision avoidance system is expected to meet three constraints:

1. Precision
2. Robustness
3. Computational efficiency

It is important to note that, if a system able to retrieve every pedestrian in a scene is desirable, it should not be to the detriment of the system precision: its ability to detect *only* pedestrians.

The perception system is intended to output the list of the pedestrians present in the environment. Every pedestrian should at least be defined by an estimate of his position in the sensing platform frame and an estimate of his velocity in a global Galilean frame. These objects features are indeed required to subsequently compute a collision risk indicator for each pedestrian.

3.3.2 Sensors choice

These criteria can in principle be met by using several sensing strategies. Indeed, any combination of a range sensor and a camera can in some way meet these requirements: the range sensor is usually not sensible to illumination and only partly sensible to weather conditions, whereas cameras allow for precise classification.

However, as mentioned in chapter 2, range sensors are not all equivalent. First, due to their low maximum range and to their limited angular resolution, sonars do not appear to be suitable for pre-crash applications. Radars could in principle be an appropriate solution because of their high maximum range and their robustness to weather conditions. Unfortunately, after some experiments, the radar that we used turned out to be not capable of generating data from pedestrians located too far away from the sensing platform as shown in figure 3.3. Consequently, although we still believe that pedestrians perception is possible with radars, we were unable to pursue our work with ours and we used a ***4-layers ladar*** instead.

Due to the high cost of infrared cameras, a visible spectrum ***monocular camera*** was chosen to complement the system.

3.3.3 Sensor combination strategy

Due to the respective characteristics of ladars and cameras, and because processing a whole camera image is usually more computationally demanding than processing a ladar scan, it seems relevant to use ladar measurements for detection, tracking and rough classification and monocular images for a refined classification of the objects released by the ladar based

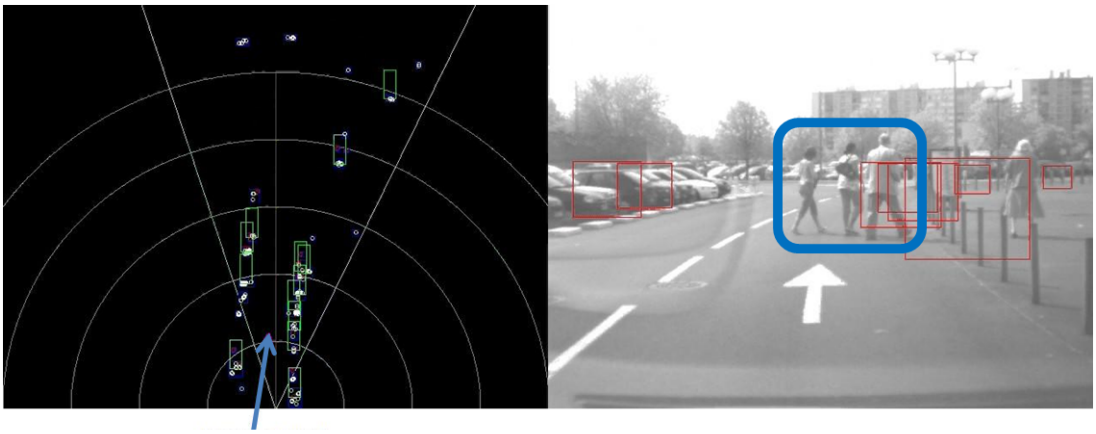


Figure 3.3: A situation where the radar do not provide any data related to the pedestrians present in the scene.

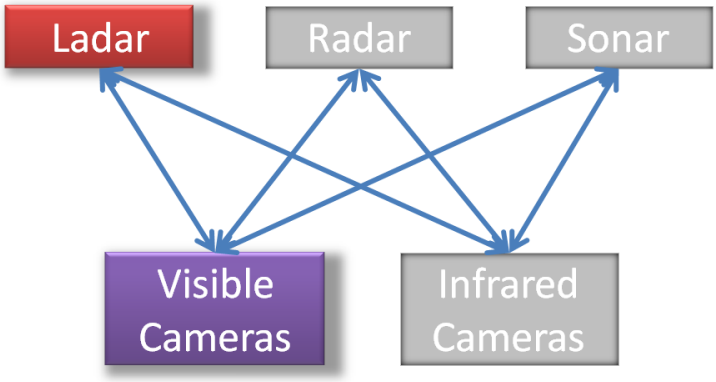


Figure 3.4: The chosen sensing strategy.

sub-system. This sensor combination strategy can be schematized as shown in figure 3.5 and summarized as follows:

- Use a ladar based sub-system to generate candidates (regions of interest in the corresponding image).
- Then use a vision based sub-system to classify the corresponding regions of interest and to filter out candidates that are not pedestrians.

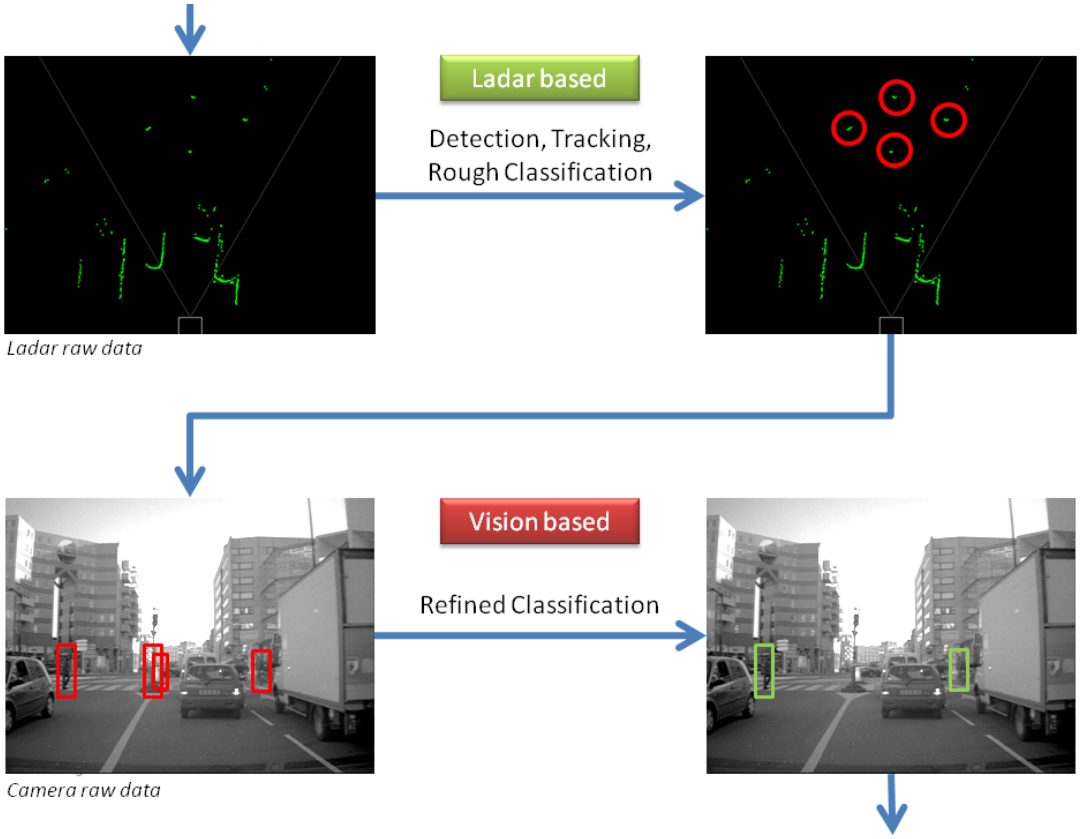


Figure 3.5: Schematic view of the combination strategy.

Three components are needed to implement this strategy:

1. A ladar based system has to be designed to produce pedestrians candidates.
2. A vision based system has to be designed to classify the regions of interest in the image related to each pedestrian candidate.
3. A method has to be designed to handle elegantly the uncertainties generated by the two sub-systems.

3.3.4 Uncertainty management

As shown in chapter 2, all the algorithms used to solve the perceptual tasks do not handle uncertainty with the same level of precision. Some of these algorithms compute posterior probability densities and hence provide a good estimate of the uncertainty involved in the result they provide. But when computational efficiency is required most usable algorithms compute estimates deprived of any uncertainty modelling. As a result, using a processing chain of three or four such fast algorithms can rapidly lead to non robust outputs.

To retain "artificially" some of these uncertainties throughout the process, we propose to maintain for each detected object a small number of additional estimates called *scores* that

will correspond to the estimated values of these three probabilities:

- **Detection score** Δ_k^i: estimate of the probability $P(D_k^i|Z_{0:k})$ with D_k^i: "the detected object i at time k is a real object".
- **Classification scores** $\Phi_k^i = (\phi_k^{i,j})_{1\leq j\leq n}$: estimates of the probabilities
$$\Big(P(C_k^{i,j}|D_k^i, Z_{0:k})\Big)_{1\leq j\leq n}$$
with $C_k^{i,j}$: "the detected object i at time k belongs to class j".
- **Tracking score** Ψ_k^i: estimate of the probability $P(T_k^i|D_k^i, Z_{0:k})$ with T_k^i: "the trajectory of the tracked object i at time k is perfectly known".

These scores will be called *uncertainty* scores in the following. In this chapter only three of the five perceptual tasks will be addressed: detection, tracking and classification. Existing approaches for *DETAC* systems are mostly based on the sequential computation of these three tasks. Maintaining these three *scores* for each initially detected objects, will allow every single sequential algorithm to potentially refine these scores. In other words, it is a simple but efficient way to authorize a certain level of interaction between sequential algorithms and to keep traces of the uncertainties generated by each of them. Besides, most intermediary hard decisions about the detected objects that are always a possible source of errors can now be postponed to the end of the process using the consolidated information aggregated into the *scores*. This uncertainty management strategy can be schematized as seen in figure 3.6.

As each algorithm can potentially refine some of these scores, methods that are used to combine the former scores with the new information brought by the current algorithm are also detailed in this chapter. The process of combining some knowledge coming from different algorithms is usually and will be in this dissertation referred to as *fusion*.

Δ_k^i, Φ_k^i and Ψ_k^i denote the estimates values at the end of the perception algorithm. Intermediary estimates of these probabilities during the process will be indicated with additional indices. For example, the final detection score Δ_k^i for object i at time k will be the result of successive incomplete estimates: $\Delta_{k,1}^i$, $\Delta_{k,2}^i$, ... ,$\Delta_{k,n}^i = \Delta_k^i$ (each one being an intermediate estimate produced along the process).

Although this approach can be applied to the detection, tracking and classification of many different types of objects at the same time (a classification score is then produced for every class of objects), only two exclusive classes ($n = 2$) will be used in this chapter : *pedestrians* and *groups of pedestrians* for reasons explained later in this chapter.

In the following sections, we present successively the ladar based algorithms used to produce pedestrians candidates, the vision based algorithms used to refine classification, and the final fusion rule used to combine the different scores estimates coming from the vision and ladar based sub-systems.

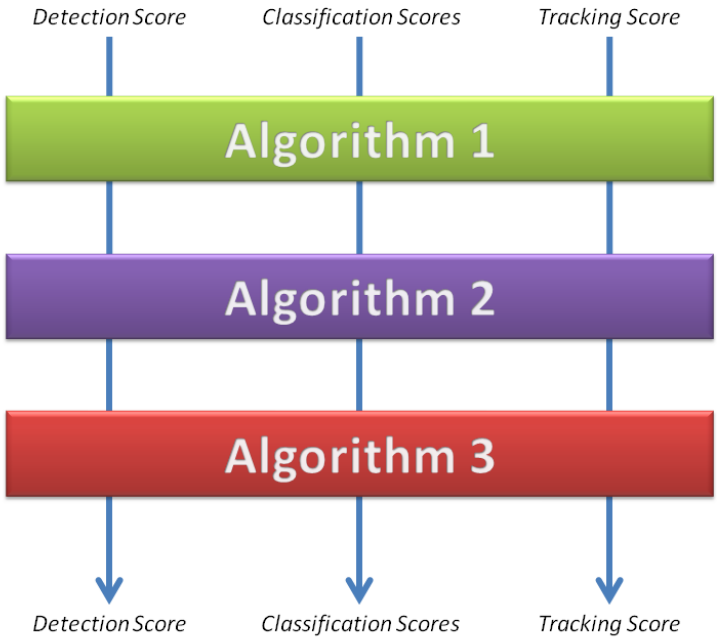


Figure 3.6: Schematic view of how uncertainties are artificially maintained throughout the process.

3.4 Ladar based algorithm

3.4.1 Principles

In this application we are primary interested in objects positions and velocities. Hence, a precise mapping and sensing platform localization is *a priori* not required. As a result, a feature-based representation of the environment is used. Every object in the environment is modelled as follows:

$$X_k^i = \begin{pmatrix} x_k^i \\ y_k^i \\ v_{x,k}^i \\ v_{y,k}^i \end{pmatrix} \tag{3.1}$$

, where x_k^i and y_k^i are the cartesian positions of the object in the sensing platform frame and $v_{x,k}^i$ and $v_{y,k}^i$ are the cartesian components of the object velocity in a Galilean referential but expressed also in the sensing platform frame. The data flow that is used is similar to most existing ladar based perception systems and can be schematized as follows.

Figure 3.7: Schematic view of the ladar sub-system data flow.

3.4.2 Objects Detection

Algorithms

The range image provided by a ladar is made of a fixed number of points that need to be grouped into objects. Ideally, all the points belonging to the same real object should be clustered together. In the absence of any dynamic information about these points, two simple geometrical observations can be used to decide if two points are likely to belong to the same object:

- **Proximity**: ladar impacts that are close are likely to belong to the same real object.
- **Alignments**: ladar impacts that are perfectly aligned are likely to belong to the same object even if they are not close.

To implement these two ideas, ladar raw data are first processed using a deterministic line fitting algorithm called *Ramer algorithm* and described in (Mendes *et al.*, 2004). This line fitting algorithm output a list of segments that is a simplified representation of the ladar raw scan. Segments contained in the list are then grouped together using the following simple distance criteria.

Two segments closer than d are grouped together

Note that the distance used between two segments is the minimum cartesian distance between two of their extremities.

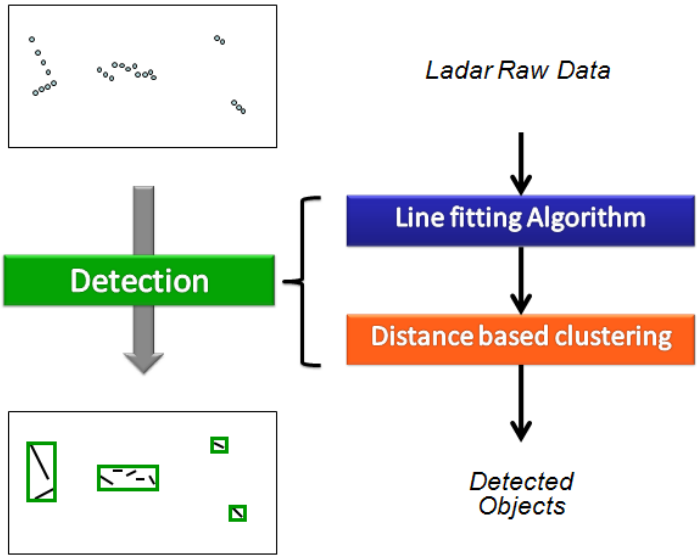


Figure 3.8: Schematic view of the ladar based detection process.

Ladar multi-layers management

The ladar used in this study features 4 sensing layers, consequently the detection algorithms described above are performed on each layer. As a result, a specific set of "detected objects" is generated on each layer. Of course, these 4 sets are strongly related to one another as a real obstacle is likely to be observed on more than one layer. This redundancy is used at this point to refine the list of detected objects that will be sent to subsequent (tracking and classification) algorithms.

We make the assumption that one of the ladar layer remains horizontal and should as such be able to collect data from all the obstacles in the scene. This layer is called the "layer of reference". For every detected obstacle in the layer of reference, the theoretical number of other layers that should observe the same obstacle is computed through simple geometrical considerations as depicted in figure 3.9. This depends of course on the range distance of the considered detected object and on the expected height of the considered obstacle. Because this system focuses on pedestrians, a height of 1.70m is used (a smaller value should be used for children detection).

Finally, a basic routine is used to filter out all the detected objects of the "reference layer" that do not appear on the expected number of other layers. The refined list of detected objects

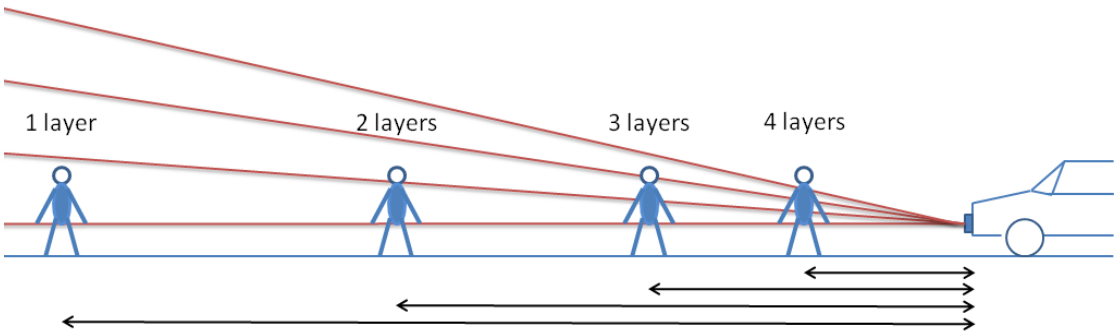


Figure 3.9: Schematic view of the number of layers that should observe a pedestrian of a given height depending on his range distance.

obtained in this way is then used in the subsequent steps of the pedestrian perception system described in this chapter without taking into account other layers anymore.

The main drawback of this multi-layer management approach is that the layer chosen as the reference layer rarely remains horizontal in practice and might occasionally miss some obstacles. A better multi-layer management strategy can certainly be designed if the vehicle pitch can be estimated (this is however not the case in our experimental setup).

Note that the whole pedestrian perception system described in this chapter is directly scalable to mono-layer ladar by simply skipping the process mentioned in this section.

The problem of groups of pedestrians

Using the above method, good detection can usually be achieved when objects are both dense (generate a high number of impacts) and not too close from each others. Unfortunately, for pedestrians these two requirements do not always hold. In urban environment, pedestrians tend to move in groups (crossings, sidewalks) and are as such often very close from each others and highly occluded. As a result, they can not always be easily discriminated from one another using a ladar. Using distance based detection algorithms, several pedestrians can be detected as one unique object that is likely to be ultimately classified as a not a pedestrian (e.g. usually for dimensions reasons) as seen in figure 3.10.

Decreasing the distance criteria d in the detection algorithm is a way to achieve better discrimination. Unfortunately this leads to incorrect detection of bigger objects. This is not satisfactory either as it can lead to a high number of false positive. A vehicle wrongly detected as two small objects can potentially generate two erroneous pedestrian candidates.

Instead of trying to discriminate every single pedestrian in a group, we propose to compute for each detected object an additional classification score related to the following object class: *groups of pedestrians*.

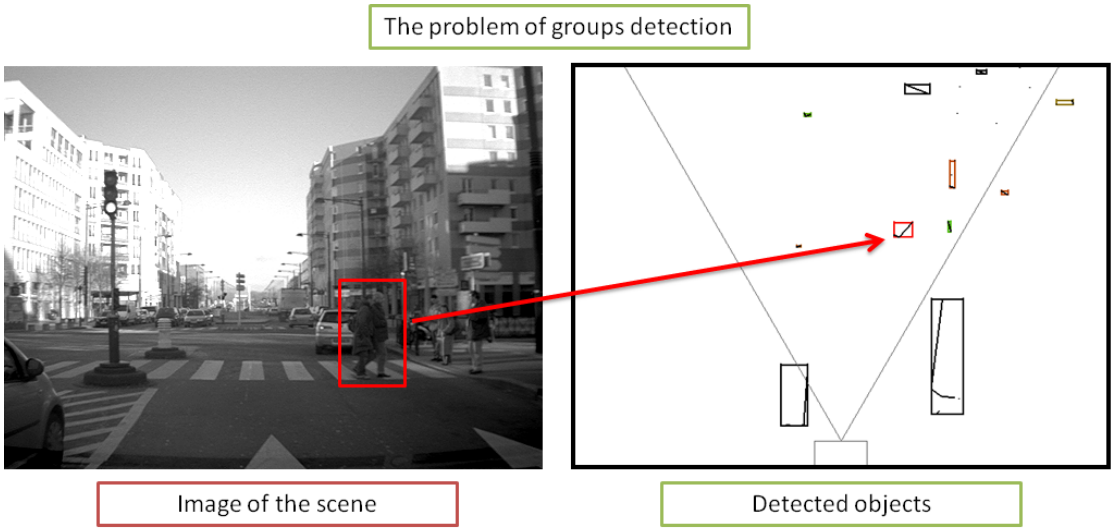


Figure 3.10: An example of the limited performances of distance based detection algorithms for groups of people detection.

First estimate of the detection probability

The detection algorithm described in the previous paragraph does not compute any uncertainty measure. As already mentioned, the goal of the *uncertainty scores* is to maintain however a fair amount of uncertainty management throughout the process. It is then necessary to compute artificially a measure of the uncertainty generated by the detection algorithm.

As bad detected objects usually contain segments that are far from each others (at distance just below the threshold d used in the detection algorithm), the following simple estimator has been proved to be relevant in practice.

$$\Delta^i_{k,1} = 1 - \frac{\text{Maximum distance between segments}}{\text{Threshold } d \text{ used for detection}} \tag{3.2}$$

First estimate of the pedestrian classification probability

After detection, the dimensions of every detected object can be estimated and first a estimation of the classification probability mass function (classification *scores*) can be made. For pedestrians, two simple ideas can be used to compute a relevant classification score:

1. A pedestrian must appear as a detected object with small dimensions, provided that he is correctly detected.

2. A pedestrian is likely to be either a totally occluded or a totally visible detected object when he is not walking in a group. This has to be understood in the sense that occluded small detected objects are likely to be parts of bigger objects.

As a consequence, a first pedestrian classification score is computed through the combination of two terms corresponding to those two heuristics.

$$\phi_{k,1}^{i,\text{pedestrian}} = \underbrace{\Gamma_1(w_k^i, l_k^i, \alpha_k^i)}_{\text{Size criterion}} \times \underbrace{\Theta(\text{occluded}_k^i)}_{\text{Occlusion criterion}} \tag{3.3}$$

, where (l_k^i, w_k^i) are the measured dimensions of the object i outlining rectangle and α_k^i the angle between its rectangle center of gravity and the x axis of the sensing platform as shown in figure 3.11.

Size criterion

The size of the every detected object is represented by the dimensions of the corresponding outlining rectangle (l_k^i, w_k^i). Based on the fact that a pedestrian can be detected as a rectangle whose dimensions can vary a lot depending on its angular position in the platform frame, the dimensions of the objects outlining rectangle should be used with care as seen in figure 3.11.

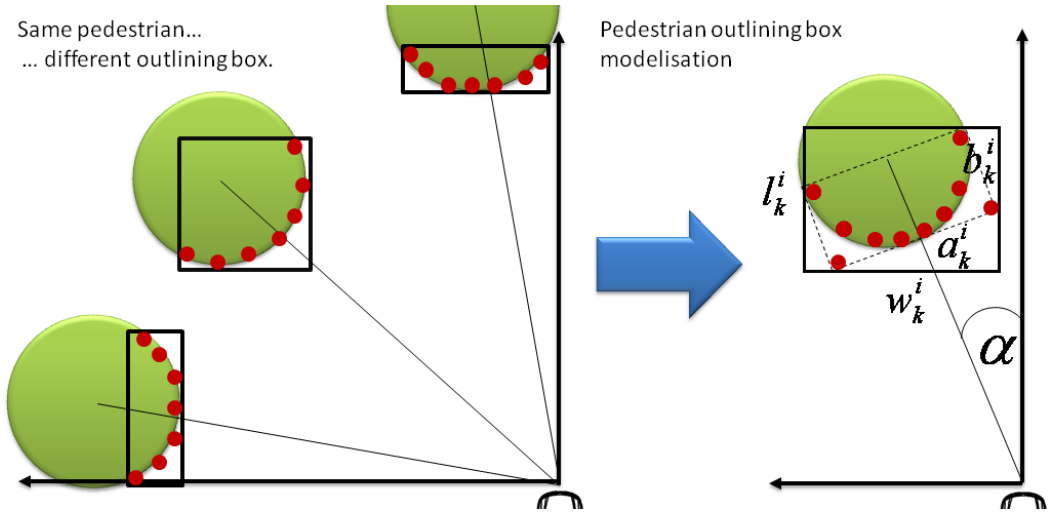


Figure 3.11: On the left: a pedestrian modelled as a cylinder can be detected as a rectangle of varying size depending on its angular position. On the right: the pedestrian model and corresponding notations.

The term $\Gamma_1(w_k^i, l_k^i)$ of equation 3.3 can be regarded as the following likelihood:

$$\Gamma_1(w_k^i, l_k^i, \alpha_k^i) = P(w_k^i, l_k^i | C_k^{i,\text{pedestrian}}, D_k^i, \alpha_k^i) \tag{3.4}$$

As for a given outlining rectangle of dimensions (w_k^i, l_k^i), there exists a unique $(\forall \alpha_k^i \neq \pi/4)$ enclosed rectangle of dimensions (a_k^i, b_k^i), this likelihood can be computed directly as follows:

$$P(w_k^i, l_k^i | C_k^{i,\text{pedestrian}}, D_k^i, \alpha_k^i) = P(a_k^i | C_k^{i,\text{pedestrian}}, D_k^i) \times P(b_k^i | C_k^{i,\text{pedestrian}}, D_k^i) \tag{3.5}$$

, where $\forall \alpha_k^i \neq \frac{\pi}{4}$,

$$a_k^i = \frac{l_k^i \cos\left|\alpha_k^i\right| - w_k^i \sin\left|\alpha_k^i\right|}{\cos\left|2\alpha_k^i\right|} \tag{3.6}$$

$$b_k^i = \frac{w_k^i \cos\left|\alpha_k^i\right| - l_k^i \sin\left|\alpha_k^i\right|}{\cos\left|2\alpha_k^i\right|} \tag{3.7}$$

It is now possible to use simple estimations of the two likelihoods exhibited in equation 3.5. In practice we measured that a pedestrian has a width between 20cm and 80cm and a visible depth that does not tap 40cm on ladar raw data. The following likelihood can then be used:

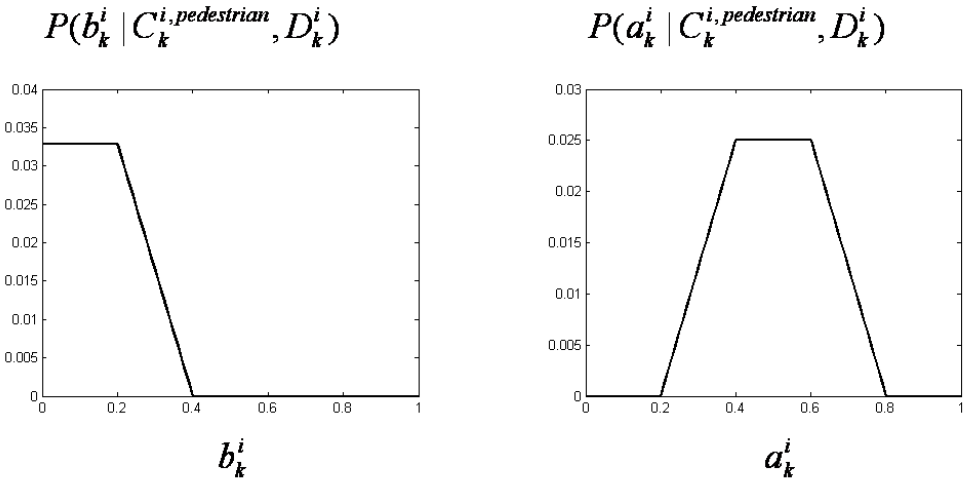
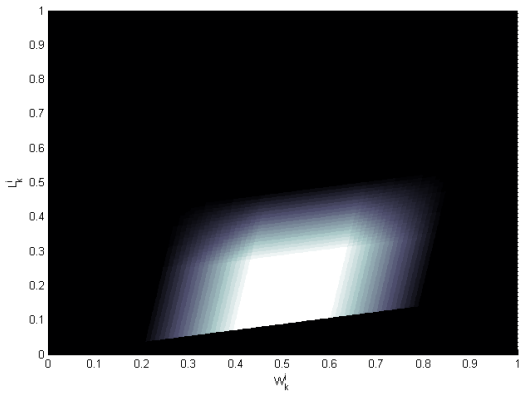
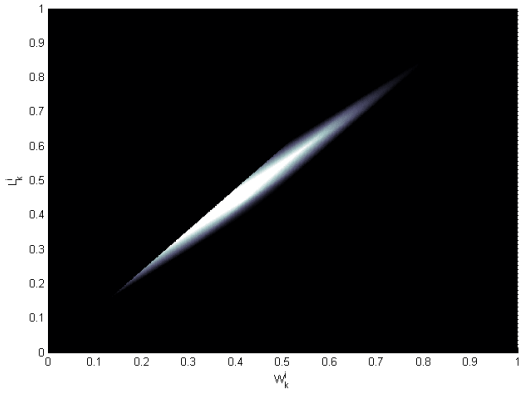


Figure 3.12: An example of two likelihoods that proved to work well in practice.

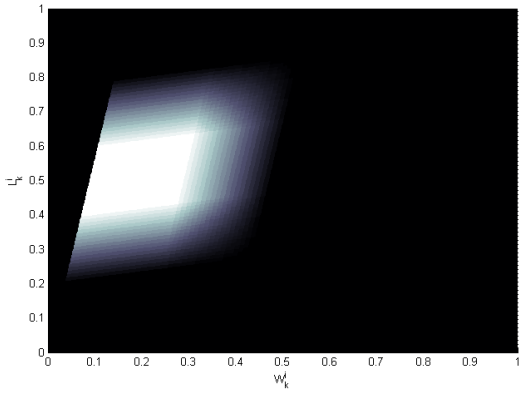
An example of the likelihood $\Gamma_1(w_k^i, l_k^i, \alpha_k^i)$ computed for three typical angular position is given in figure 3.13. These results are consistent with the fact that a pedestrian in front of the vehicle ($\alpha_k^i = 10$) should be detected as a horizontal rectangle, a pedestrian on the side of the vehicle ($\alpha_k^i = 80$) should appear as a vertical rectangle and a pedestrian located in between these two extremes ($\alpha_k^i = 50$) should be detected almost as a square.



(a) $\alpha_k^i = 10$ degrees



(b) $\alpha_k^i = 50$ degrees



(c) $\alpha_k^i = 80$ degrees

Figure 3.13: Example of $\Gamma_1(w_k^i, l_k^i, \alpha_k^i)$ for different value of α_k^i.

Occlusions criterion

As mentioned above, in practice, single pedestrians are rarely *partially* occluded. They can of course be occluded but they are then either totally invisible to sensor measurements or in a group (that case is addressed later in this section). As a result, the term Θ is used to penalize objects that are partially occluded.

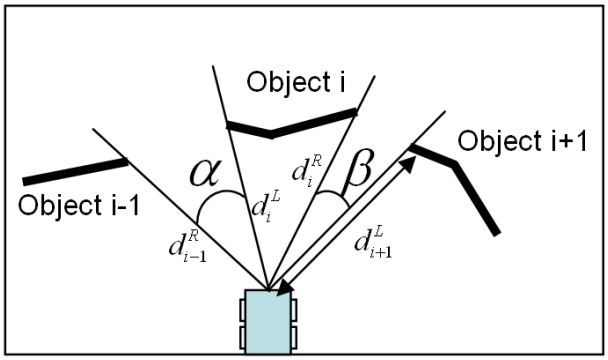


Figure 3.14: Object i is labelled as occluded depending on the relative position of objects $i-1$ and objects $i+1$ using the rule below.

That is why, using the notation of figure 3.14, each detected object i at time k is labeled either as "occluded" or "not occluded" following the rule:

$$\text{occluded}_k^i = \begin{cases} 1 & \text{if } (\alpha < \text{resolution}) \wedge (d_{i-1}^R < d_i^L) \ \vee \ (\beta < \text{resolution}) \wedge (d_i^R > d_{i+1}^L) = 1 \\ 0 & \text{else} \end{cases} \tag{3.8}$$

The term *resolution* refers to the angular horizontal resolution of the ladar used (this resolution is for example equal to 0.5 degree in our experiments). Then, the term Θ is given a binary value as follows:

$$\Theta(\text{occluded}_k^i) = \begin{cases} 1 & \text{if occluded}_k^i = 0 \\ 0 & \text{else} \end{cases} \tag{3.9}$$

First estimate of the group classification probability

If some of the previous classification principles might apply to a group of people, the size criterion need of course to be altered and an additional criteria should be used to differentiate for example groups of people from all the vehicles that are usually present in urban scenes. Groups of people usually appear as relatively big obstacles made of a high number of unordered segments. On the contrary, many other obstacles of similar size are composed of highly structured segments configuration.

The segment configuration of a detected object i at time k is noted as follows:

$$\text{seg}_k^i = \left\{ s_{k,1}^i, s_{k,2}^i, ..., s_{k,n}^i \right\}$$

, where $(s^i_{k,j})_{1\leq j\leq n}$ are the segments that have been produced and grouped together by detection algorithms on the horizontal ladar layer (layer of reference).
A good first estimate of the groups classification probability can be computed using the following combination:

$$\phi^{i,\text{groups}}_{k,1} = \underbrace{\Delta(\text{seg}^i_k)}_{\text{Segment configuration criterion}} \times \underbrace{\Gamma_2(w^i_k, l^i_k)}_{\text{Size criterion}} \times \underbrace{\Theta(\text{occluded}^i_k)}_{\text{Occlusion criterion}} \tag{3.10}$$

While the last term of the above equation remains identical to the one used in equation 3.3, the first two terms need to be explained:

Size criterion

A group can include several pedestrians and the method described above to compute the likelihood that a detected object with an outlining rectangle of given dimensions is a pedestrian do not scale well to groups.

In fact, the dimensions of such obstacles do not have any upper bound. A group of 100 people can indeed be detected as a very big obstacle. However, these dimensions are unlikely to be lower than a specific set of values $(w_{\min}, l_{\min})$ that depend on the angular position of the group. We propose to build a simple likelihood function from the estimation of these minimum dimensions.

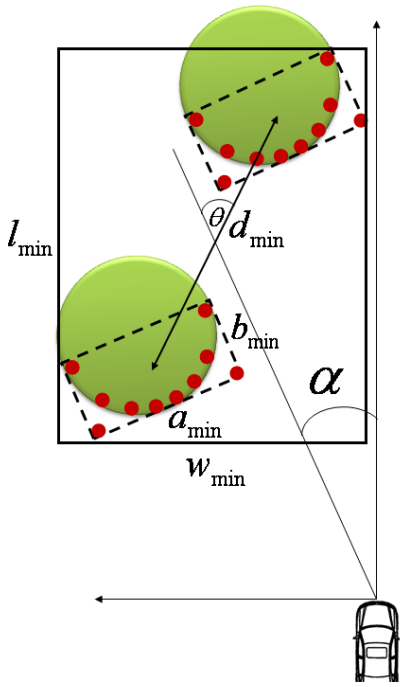


Figure 3.15: The group model used to compute the minimum dimensions that such obstacles should have depending on α.

We assume that the smallest group is made of two pedestrians at a distance $d_{\min}$ as shown in figure 3.15. The minimum dimensions of the resulting outlining box are given by:

$$w_{\min} = a_{\min}\cos\alpha + b_{\min}\sin\alpha + d_{\min}\,|\sin(\theta-\alpha)| \tag{3.11}$$

$$l_{\min} = a_{\min}\sin\alpha + b_{\min}\cos\alpha + d_{\min}\,|\cos(\theta-\alpha)| \tag{3.12}$$

As a consequence, for a given object angular position α_k^i, the dimensions (w_k^i, l_k^i) of the outlining box should meet the following requirements:

$$\begin{cases} w_k^i & \geq w_{\min} \\ l_k^i & \geq l_{\min} \end{cases} \quad \Longrightarrow \quad \begin{cases} (w_k^i - A)^2 + (l_k^i - B)^2 & \geq d_{min}^2 \\ w_k^i & \geq A \\ l_k^i & \geq B \end{cases} \tag{3.13}$$

, with $A = a_{\min}\cos\alpha_k^i + b_{\min}\sin\alpha_k^i$ and $B = a_{\min}\sin\alpha_k^i + b_{\min}\cos\alpha_k^i$.

The above algebraic equations only define the minimum dimensions of a group of people as a function of α. A simple likelihood $\Gamma_2(w_k^i, l_k^i, \alpha_k^i)$ can be obtained by forcing $\Gamma_2(w_k^i, l_k^i, \alpha_k^i) = 1$ when $w_k^i > w_{\min}$ or $l_k^i > l_{\min}$ and $\Gamma_2(w_k^i, l_k^i, \alpha_k^i) = 0$ in any other situation. The transition between these two states is made linear as shown in figure 3.16.

$$\Gamma_2(w_k^i, l_k^i, \alpha_k^i) = P(w_k^i, l_k^i | C_k^{i,\text{groups}}, D_k^i, \alpha_k^i) \tag{3.14}$$

The examples shown in figure 3.16 are consistent with the fact that the smallest possible group will be seen as a horizontal rectangle when α_k^i is small and as a vertical rectangle when α_k^i is bigger.

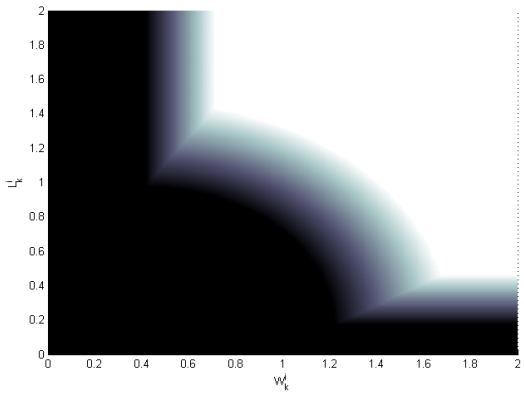
Segments configuration criterion

In order to penalize big obstacles that are not groups of people, the following observation is used: groups of people tend to be composed of unordered small segments as opposed to other big obstacles (vehicles, walls, bus, etc...) that tend to be made of a few number of big segments.

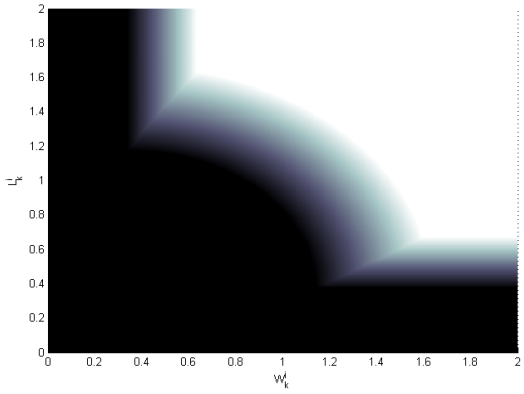
$$\Delta(seg_k^i) = \prod_{1\leq j\leq n} f(\|s_{k,j}^i\|) \tag{3.15}$$

where f is a trapezoidal function that penalizes both small and very large segments:

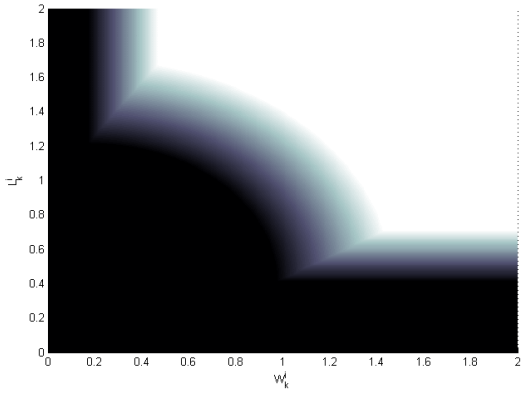
$$f(x) = \begin{cases} 0 & \forall x < a_1, x \geq a_4 \\ \frac{x-a_1}{a_2-a_1} & \forall a_1 \leq x < a_2 \\ 1 & \forall a_2 \leq x < a_3 \\ \frac{a_4-x}{a_4-a_3} & \forall a_3 \leq x < a_4 \end{cases} \tag{3.16}$$



(a) $\alpha_k^i = 10$ degrees



(b) $\alpha_k^i = 50$ degrees



(c) $\alpha_k^i = 80$ degrees

Figure 3.16: Example of $\Gamma_2(w_k^i, l_k^i, \alpha_k^i)$ for different values of α_k^i.

In our experiments, the following values were chosen:

$$(a_1, a_2, a_3, a_4) = (0.2m, 0.4m, 0.6m, 0.8m)$$

This simple criterion proved to be surprisingly efficient in discriminating groups of people from other "big" obstacles as shown later in this chapter.

Conclusion

By discriminating all the objects in the scene and providing rich geometrical information about them, the detection algorithms used allow to compute relevant initial estimates for the detection and classification probabilities as schematized in figure 3.17.

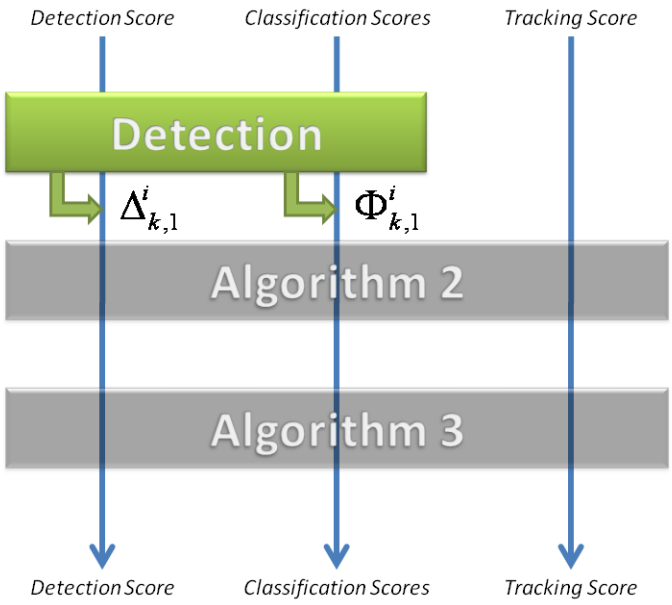


Figure 3.17: The detection algorithms alter the detection and classification scores.

3.4.3 Objects Tracking

Algorithms

Data association

As mentioned above, detection algorithms can produce erroneous obstacles: a real object detected as one object at time t can unfortunately be detected as two separate objects at time $t+1$. As a consequence, the association of more than one detected object to one tracked object should be made possible. This can be easily implemented using the *joint probabilistic data association approach (JPDA)* detailed in chapter 2. Every tracked object is associated with all the currently detected objects that are in its vicinity (distance $\leq \alpha$) and an association probability $\beta_{i,j}$ is computed for each possible association between tracked object i and the currently detected object j whose center of gravity coordinates are noted X_k^j.

$$\beta_{i,j} \propto \underbrace{\mathcal{N}(X_k^j, \mu_k^i, \Sigma_k^i)}_{\text{Mahalanobis distance}} \times \underbrace{\Delta_{k,1}^j}_{\text{Detection score}}$$

The Mahalanobis distance is commonly used in the computation of such association probabilities. But the additional information contained in the detection score $\Delta_{k,1}^j$ proved to be relevant in practice as it decreases the influence in the filtering process of the detected objects that are likely to be erroneous.

Filtering

Assuming that the density probability of the random state vector $(x_k^i, y_k^i, v_{x,k}^i, v_{y,k}^i)$ and the noises of in the motion and measurement model are Gaussian, Kalman filters are used to recursively estimate the tracked objects state vectors from all the weighted association hypothesis. In practice a constant velocity motion model is used.

Note that to be useful for subsequent classification probabilities estimations, the velocity of each tracked object is estimated in a global and Galilean reference frame. Doing so requires additional data related to the state of the sensing vehicle (vehicle speed and orientation). These data can be inferred from the perception sensors (localization task) or can be directly measured through inertial sensors on the sensing platform. In this chapter, the localization of the sensing platform is directly inferred from proprioceptive measurements. The tracking process can be schematized as shown in figure 3.18.

Tracked objects creation and deletion

Every time a new objet is detected and not associated with any previously seen tracked objects, a new tracker is initialized on that object. When a tracked object is not seen anymore in the data, it remains tracked until the trace of its state covariance matrix (estimated through Kalman filtering) reaches a certain value.

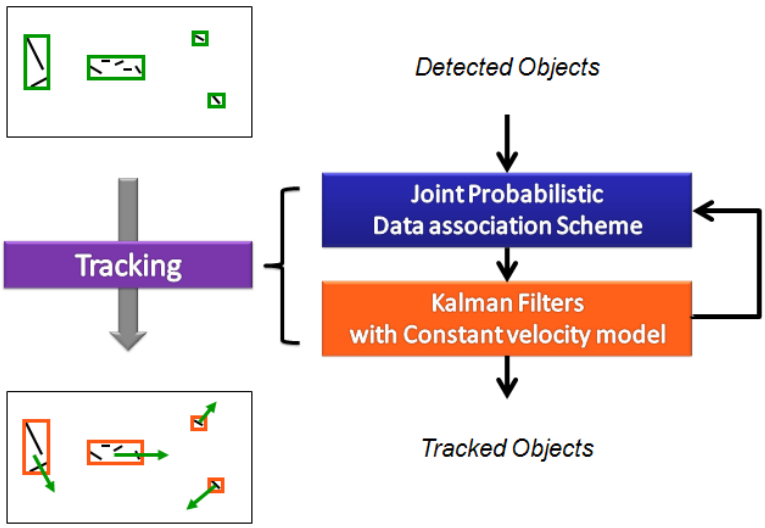


Figure 3.18: Schematic view of the ladar based tracking process.

First estimation of the tracking probability

The Kalman filter based filtering approach offers a relevant measure of the confidence that is placed in tracking accuracy. Indeed, the covariance P_k^i of the state estimate X_k^i of object i contain information about the uncertainties related to the state estimate. Based on this covariance matrix, we compute a scalar tracking score as follows:

$$\Psi_{k,2}^i = \sqrt{\frac{\text{trace}(P_{\text{final}})}{\text{trace}(P_k^i)}} \tag{3.17}$$

, where P_{final} is the theoretical value of P_k^i after convergence of the Kalman filter. For simplicity, P_{final} is estimated experimentally (the noise matrices in the Kalman equations are then set to a fixed value and should not be modified without resetting the value of P_{final}).

Refined estimation of the detection and classification probabilities

Initial classification scores have already been computed by the detection algorithms. However, these estimations can be refined in two ways using the tracking algorithms. First, velocity estimates are now available and can help to penalise non human obstacles (when velocities are too high). Second, tracking can be used to smooth these scores over time.

As a human obstacle should have a bounded velocity the following term is used:

$$\Omega(v_k^i) = \begin{cases} 1 & \text{if } v_k^i \leq v_{\max} \\ 0 & \text{else} \end{cases} \tag{3.18}$$

Then the filtering strategy based on the *JPDA* association scheme is applied to the detection and classifications scores as follows:

$$\Delta^i_{k,2} \propto \underbrace{\Delta^i_{k-1,2}}_{\text{Prior}} \sum_j \underbrace{\Delta^j_{k,1}}_{\text{Conditional Likelihood}} \times \underbrace{\beta_{i,j}}_{\text{Association probability}} \tag{3.19}$$

For the classification scores, the velocity criterion is added in the filtering equations:

$$\Phi^{i,\text{pedestrian}}_{k,2} \propto \Omega(v^i_k) \times \Phi^{i,\text{pedestrian}}_{k-1,2} \sum_j \Phi^{j,\text{pedestrian}}_{k,1} \times \beta_{i,j} \tag{3.20}$$

$$\Phi^{i,\text{groups}}_{k,2} \propto \Omega(v^i_k) \times \Phi^{i,\text{groups}}_{k-1,2} \sum_j \Phi^{j,\text{groups}}_{k,1} \times \beta_{i,j} \tag{3.21}$$

Conclusion

By solving the data association problem and estimating tracked objects velocities, the tracking algorithms allow to refine existing detection and classification scores and to compute a first relevant tracking score. It is also an efficient way to compute smooth estimates of object positions in the environment. Uncertainty management at this point of the process can be schematized as shown in figure 3.19.

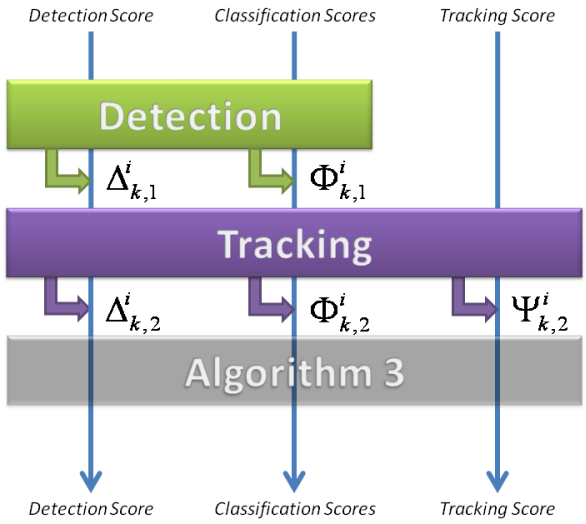


Figure 3.19: The tracking algorithms alter the detection, classification and tracking scores.

3.4.4 Objects Rough Classification

Algorithms

To avoid any anticipated decisions, no objects are discarded during the detection and tracking processes. In other words, all detected objects are tracked. While maintaining a high number of tracked objects in our features-based representation is not prohibitive in term of computational requirements, processing a great number of regions of interest in the monocular image is very demanding.

To avoid unnecessary computations of the vision based algorithms, tracked objects that are unlikely to be pedestrians or groups of people are discarded.

This screening process is naturally based on the information successively aggregated in the *uncertainty scores*. In the context of our experiments whose results are presented later in this chapter, only the tracked obstacles satisfying the following thresholds are projected into the corresponding monocular image and sent to the vision based algorithm.

Detection score	$\Delta^i_{k,2}$	≥ 0.3
Classification score (pedestrian or group)	$\Phi^{i,j}_{k,2}$	≥ 0.6
Tracking score	$\Psi^i_{k,2}$	≥ 0.5

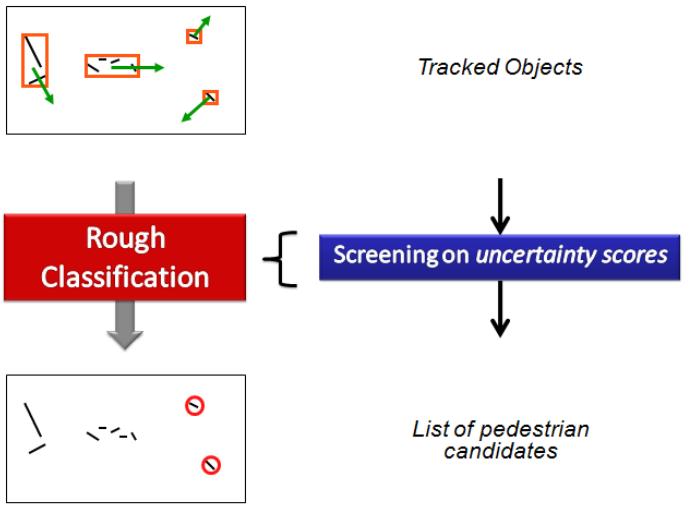


Figure 3.20: Schematic view of the ladar based rough classification process.

Scores handling

This simple classification algorithm do not alter any *uncertainty scores*. However for clarity, at this point the current *uncertainty scores* are indexed with the word "ladar" as they represent the last scores estimated by the ladar based algorithms as seen in figure 3.21.

Conclusion

The rough classification operated by the screening algorithm presented above is mostly based on the uncertainty scores computed in the previous phases of the process. In that sense, the overall classification of the detected objects is spread in all the successive perception algorithms used in the ladar sub-system. An overview of the ladar based perception sub-system is shown in figure 3.21.

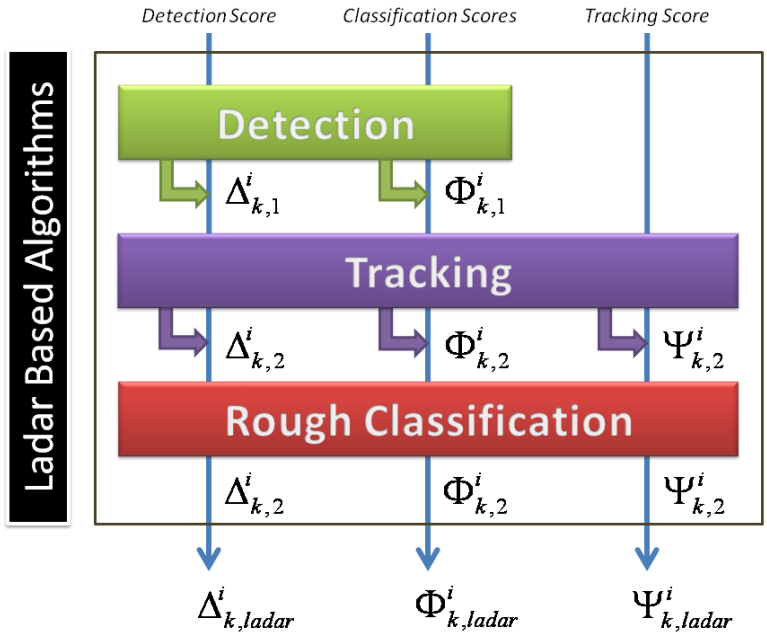


Figure 3.21: Schematic view of the uncertainty management within the ladar based perception sub-system.

3.5 Vision based classification algorithm

3.5.1 Camera Image Projection

Every pedestrian or group candidate sent by the ladar based perception sub-system is projected in the corresponding (closest in time) calibrated camera image. This projection is performed assuming that the sensing platform is neither rolling nor pitching and using the flat world assumption.

This projection is of course dependent on a precise calibration of the intrinsic parameters of the camera (focal, distortion parameters...) and on a correct estimation of the camera location compared to the location of the ladar. These tasks have been performed with the help of the calibration routines available in the *OpenCV*[8] package.

Each candidate noted $X^i_{k,\text{ladar}}$ sent by the ladar based sub-system produces a region of interest $X^i_{k,\text{camera}}$ in the corresponding image. As the height of pedestrians and groups cannot be known precisely from the ladar data, this parameter is set to a standard value. In practice, a significant number of candidates are also out of the camera field of view after projection. These invisible objects on the monocular image are not discarded but will not be further classified by the vision based sub-system. Regions of interest that are in the camera field and processed by the algorithm described below.

3.5.2 A Boosting classification based approach: AdaBoost

The AdaBoost algorithm, introduced in 1995 by Y. Freund and R. Shapire (Freund & Schapire, 1995; Freund & Schapire, 1999) is based on the idea that a series of trained *weak classifiers* can build a strong and efficient classifier as depicted in figure 3.22.

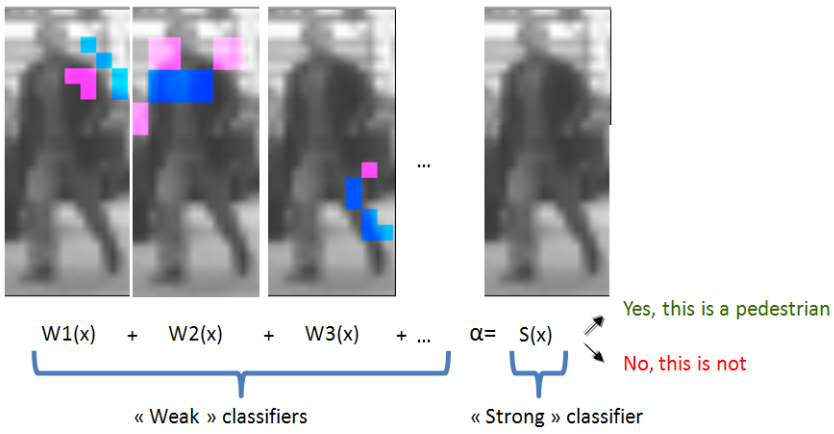


Figure 3.22: Principle of Boosting.

Viola and Jones proposed in (Viola & Jones, 2001; Viola *et al.*, 2003) weak classifiers based on Haar-like features that provide interesting face and pedestrian detection capabilities. More

[8]OpenCV (Open Source Computer Vision) is a library of programming functions for real time computer vision under BSD license. http://opencv.willowgarage.com

recent works introduced novel types of features that proved to be even more efficient such as the "control points" (Abramson *et al.*, 2007) and the "connected control points" (Stanciulescu *et al.*, 2007). A complete overview of monocular approaches for pedestrian detection can be found in (Enzweiler & Gavrila, 2009). The former type of features allows both faster computation (140000 36pixels $\times$ 36pixels images per second on a 2.5GHz core) and better performance as shown in (Stanciulescu *et al.*, 2007).

The vision based classification algorithm used here is based on 500 connected control points features that were learned using a genetic algorithm on a database of 4800 positive and 8400 negative images. Positive images were labelled by hand but through a specific software designed to assist the user in this repetitive task.

Two problems have to be handled. First, because the robot can roll or pitch, there can be an offset between the position of the projected region and the real obstacle. And second, groups candidates can produce large regions of interest with several pedestrians in the image while the vision based algorithm described above is specifically and only intended to classify a region as being a pedestrian or not. To overcome these two difficulties, all the relevant zones that are likely to contain a single pedestrian around or within the projected region is processed by the algorithm.

To accelerate the search of the vision-based classification algorithm inside the large regions that are generated by "groups"candidates, we experimented to use before image projection a simple distance based clustering algorithm (detection algorithm) to discriminate roughly where individual pedestrians might be located inside these detected objects classified as groups. While not being very precise, this method allows for smaller projected image regions and accelerates the vision-based algorithm process.

In both cases, the projected region $X^i_{k,\text{camera}}$ is decomposed into n different regions:

$$(X^{i,p}_{k,\text{camera}})_{1\leq p\leq n}$$

Each one of these n regions is processed by the vision based classification algorithm and is given a voting value $\Upsilon^{i,p}_k$ corresponding to the sum of the weighted combination of the 500 weak classifiers $(w_l)_{1\leq l\leq 500}$.

$$\Upsilon^{i,p}_k = \sum_{1\leq l\leq 500} \alpha_l w_l(X^{i,p}_{k,\text{camera}}) \tag{3.22}$$

Finally, for each projected image $X^i_{k,\text{camera}}$ a global classification score is computed as follows:

$$\Phi^{i,\text{pedestrian}}_{k,\text{camera}} = \Phi^{i,\text{group}}_{k,\text{camera}} = \max_{1\leq p\leq n} \Upsilon^{i,p}_k \qquad \forall \text{ regions inside camera } FoV \tag{3.23}$$

Pedestrian or group candidates that are out of the camera field of view (FoV) are given a non informative camera based classification score:

$$\Phi^{i,\text{group}}_{k,\text{camera}} = \Phi^{i,\text{group}}_{k,\text{camera}} = 0.5 \qquad \forall \text{ regions outside camera } FoV \tag{3.24}$$

3.5.3 Uncertainty management

As the vision based algorithm described above is only trained to compute a classification probability for a given region to be a pedestrian, no detection or tracking information can be deduced from such a result. However, as explained above every projected region in the image is given a new classification scores $\Phi^i_{k,\text{camera}}$.

For any given object i at time k, these vision based classification scores are of course independent and different from the uncertainty scores already computed by the ladar based sub-system as depicted in figure 3.23. The fusion rule used to combine those different scores is detailed in the next section.

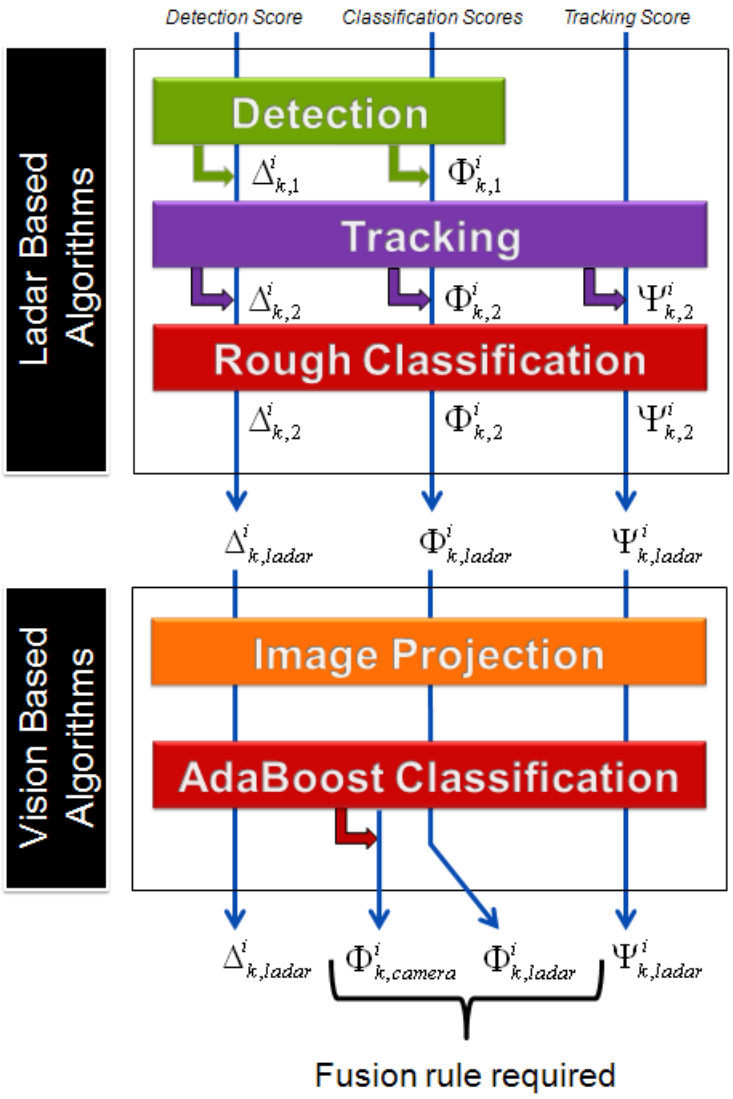


Figure 3.23: Schematic view of the *uncertainty scores* computed by the two sub-systems.

3.6 Final Fusion Rule

3.6.1 Principle

Data Fusion is an active scientific research area where lots of approaches based on different mathematical formalisms have been proposed over the years. The theory of possibilities (Dubois, 1985) and the belief theory (or Dempster-Shafer theory) (Dempster, 2008; Shafer, 1976) have enjoyed a growing popularity in a recent past but the more common probability theory is still of wide use. If using the possibility or belief theories can undoubtedly lead to efficient fusion algorithms, the deployment of a sophisticated fusion scheme was not the main objective here. In fact, the simple probability based fusion rule detailed below turned out to be quite efficient in practice.

As shown in figure 3.23, two different estimates of the same probabilities $P(C_k^{i,j}|D_k^i, Z_{0:k})$ have to be combined with $j \in \{\text{pedestrian, group}\}$. These estimates have been computed by algorithms that are of course subject to uncertainties. As a consequence, each estimate can be rewritten as follows:

$$P(C_k^{i,j}|D_k^i, A_k) = \Phi_{k,\text{ladar}}^{i,j} \tag{3.25}$$

$$P(C_k^{i,j}|D_k^i, B_k) = \Phi_{k,\text{camera}}^{i,j} \tag{3.26}$$

, where A_k: "Ladar based algorithms have produced the perfect estimate given $Z_{0:k}$" and B_k: "Vision based algorithms have produced the perfect estimate given $Z_{0:k}$". Assuming that A_k and B_k are independent, the following fusion rule can be derived from basic Probability rules:

$$\begin{aligned} P(C_k^{i,j}|D_k^i, Z_{0:k}) \simeq\ & P(C_k^{i,j}|D_k^i, A_k)P(A_k)P(\overline{B_k}) + P(C_k^{i,j}|D_k^i, B_k)P(\overline{A_k})P(B_k) \\ & + P(C_k^{i,j}|D_k^i, A_k, B_k)P(A_k)P(B_k) + P(C_k^{i,j}|D_k^i, \overline{A_k}, \overline{B_k}, Z_{0:k})P(\overline{A_k})P(\overline{B_k}) \end{aligned} \tag{3.27}$$

, where

$$\begin{array}{ll} P(C_k^{i,j}|D_k^i, A_k) & \simeq \Phi_{k,\text{ladar}}^{i,j} \\ P(C_k^{i,j}|D_k^i, B_k) & \simeq \Phi_{k,\text{camera}}^{i,j} \\ P(C_k^{i,j}|D_k^i, A_k, B_k) & \simeq (\Phi_{k,\text{ladar}}^{i,j}P(A_k) + \Phi_{k,\text{camera}}^{i,j}P(B_k))(P(A_k) + P(B_k))^{-1} \\ P(C_k^{i,j}|D_k^i, \overline{A_k}, \overline{B_k}, Z_{0:k}) & \simeq \epsilon \quad (a\ priori \text{ classification probabilities}) \end{array} \tag{3.28}$$

This fusion rule can then be written as follows:

$$\begin{aligned} \Phi_{k,\text{final}}^{i,j} = & \Phi_{k,\text{ladar}}^{i,j}P(A_k)P(\overline{B_k}) + \Phi_{k,\text{camera}}^{i,j}P(\overline{A_k})P(B_k) \\ & + \frac{\Phi_{k,\text{ladar}}^{i,j}P(A_k) + \Phi_{k,\text{camera}}^{i,j}P(B_k))}{P(A_k) + P(B_k)}P(A_k)P(B_k) + \epsilon P(\overline{A_k})P(\overline{B_k}) \end{aligned} \tag{3.29}$$

In our experiments, whose results are presented in the next section, the a priori classification probabilities was set to 0.1 for both pedestrians and groups, the confidence placed in the ability of the ladar based sub-system to produce a correct estimate was set to 0.4 and the similar confidence related to the vision based sub-system was set to 0.8. This fusion rule can be schematized as shown in figure 3.24.

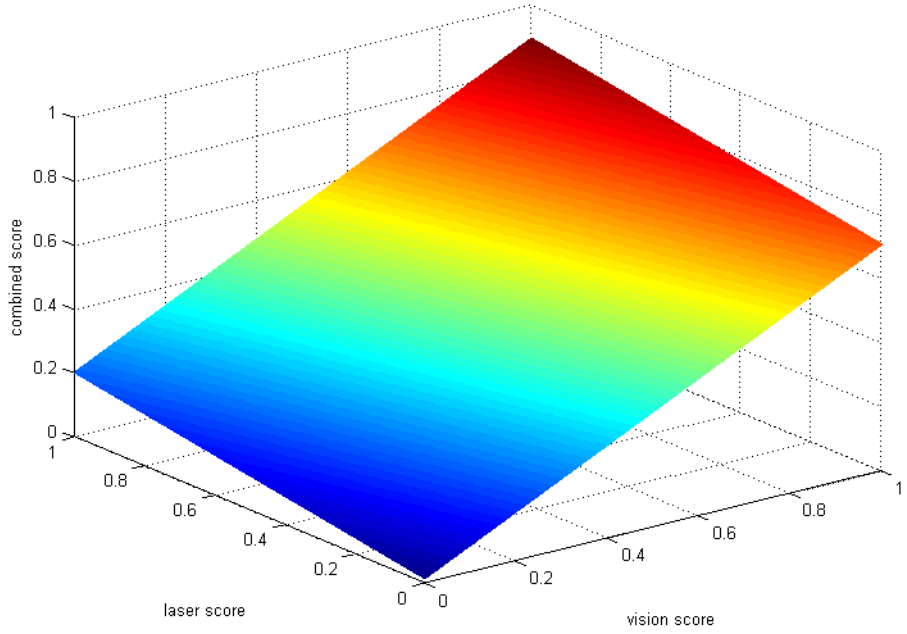


Figure 3.24: Schematic view of the fusion rule with $P(A_k) = 0.4$, $P(B_k) = 0.8$ and $P(C_k^{i,j}|D_k^i, \overline{A_k}, \overline{B_k}, Z_{0:k}) = 0.1$.

3.6.2 Uncertainty management

It is important to note that only the classification scores are combined using the above fusion rules. The detection and tracking scores computed by the ladar based sub-system are not altered neither by the vision based sub-system nor by the fusion scheme. At this point of the process, final estimates for the detection, classification and tracking probabilities are available.

$$\begin{array}{ll} \Delta^i_{k,\text{final}} & = \Delta^i_{k,\text{ladar}} \\ \Phi^{i,j}_{k,\text{final}} & = f_{\text{fusion rule}}(\Phi^{i,j}_{k,\text{ladar}}, \Phi^{i,j}_{k,\text{camera}}) \\ \Psi^i_{k,\text{final}} & = \Psi^i_{k,\text{ladar}} \end{array} \tag{3.30}$$

3.6.3 Conclusion

The list of objects that is now available is similar to the one provided by the ladar sub-system $(X^i_{k,\text{final}})_{1\leq i\leq N} = (X^i_{k,\text{ladar}})_{1\leq i\leq N}$ with however corresponding scores that have been consolidated by the fusion process. An appropriate screening can now be performed on these scores to retain only the relevant pedestrians and groups. A global view of the perception system described in this chapter is given in figure 3.25.

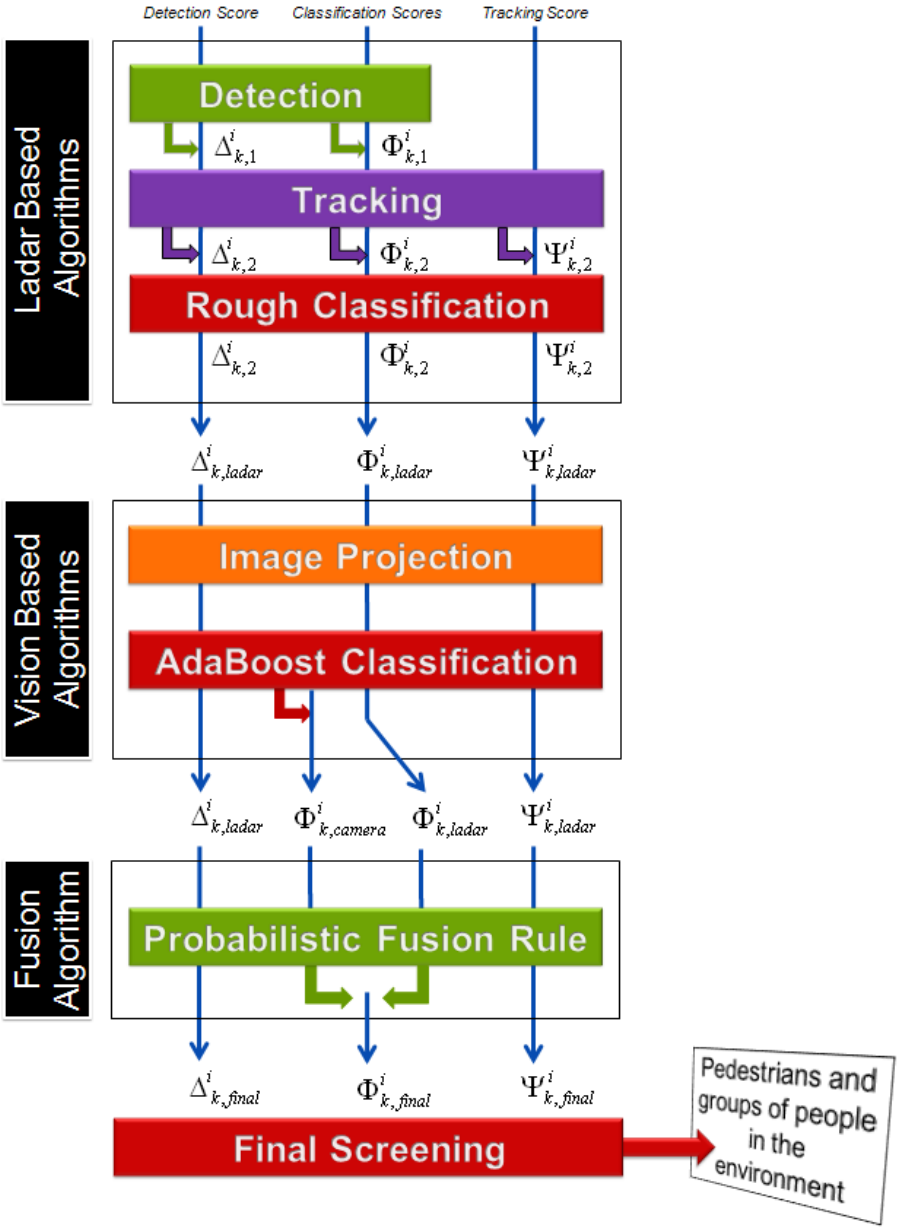


Figure 3.25: Global view of the perception system proposed in this chapter.

3.7 Experiments

3.7.1 Experimental setup

The results presented in this section have been obtained from real data recorded in various urban environments in the context of the french national research project *LOVe* mentioned in the introduction of this dissertation. An overview of the specifications of the monocular camera and of the ladar that have been used is given below.

Ladar Ibeo Alasca XT

Number of layers	4
Horizontal angular resolution	0.5 degree
Vertical angular resolution	0.8 degree
Maximum range	200 meters
Field of view	150 degrees
Acquisition rate	10Hz

Camera Cypress Smal

Type	Black and White
Resolution	480×640 pixels
Acquisition rate	30 images per seconds
Digital *SNR*	45 dB

Proprioceptive sensors

Vehicle Speed	Odometers through *CAN* bus
Vehicle Yaw rate	*IMU* Gyroscopes through *CAN* bus

Figure 3.26: Overview of the ladar used in the experiments presented in this section.

3.7.2 Methods of evaluation

To quantitatively assess the benefit of our approach, we constructed manually a ground truth representing every single pedestrian that a perfect pedestrian perception system would detect. The output of our algorithm is then automatically compared to the perfect detector output on each camera frame. The usual "Precision" and "Recall" parameters are used to quantify the performances. Because only pedestrians have been labelled on data sequences, it is not directly possible to assess the capabilities of the system as a "pure" obstacle detector. In the same manner, the tracking performances are not evaluated here.

$$\text{Precision} = \frac{\text{Valid pedestrians given by the algorithm}}{\text{All pedestrians given by the algorithm}}$$

$$\text{Recall} = \frac{\text{Valid pedestrians given by the algorithm}}{\text{All pedestrians present in the scene}}$$

Several couples (Precision, Recall) can of course be obtained depending on the set of thresholds $(\alpha_{\text{detection}}, \alpha_{\text{pedestrian}}, \alpha_{\text{group}}, \alpha_{\text{tracking}})$ applied to the corresponding *uncertainty scores* that are chosen to finally decide which objects should be released by the system.

3.7.3 Optimisation procedure

Before presenting any results, a brief overview of the procedure followed to set some of the above threshold is given in this paragraph. Indeed, the influence of the four thresholds cannot be easily analysed without setting some of them beforehand.

The influence of the detection and tracking thresholds have first been analysed with fixed values for the classification scores. It appeared that $\alpha_{\text{detection}} = 0.2$ and $\alpha_{\text{tracking}} = 0.5$ are both values that work well in practice.

Then with these two specific values, the influence of $\alpha_{\text{pedestrian}}$ and α_{group} were analysed through a set of *Precision VS Recall* curves shown in figure 3.27. It was thus possible to define a simple staircase function g to drive these two thresholds at the same time from one single global threshold $\alpha_{\text{classification}}$.

$$(\alpha_{\text{pedestrian}}, \alpha_{\text{group}}) = g(\alpha_{\text{classification}})$$

The main objective of this function is to allow a quick tuning of the system through a single parameter in order to adapt its performances to specific situations. In the remaining of this chapter, unless stated otherwise *Precision VS Recall* curves are obtained by successively modifying the setting of this global classification threshold, all other thresholds being set to the values detailed in this paragraph.

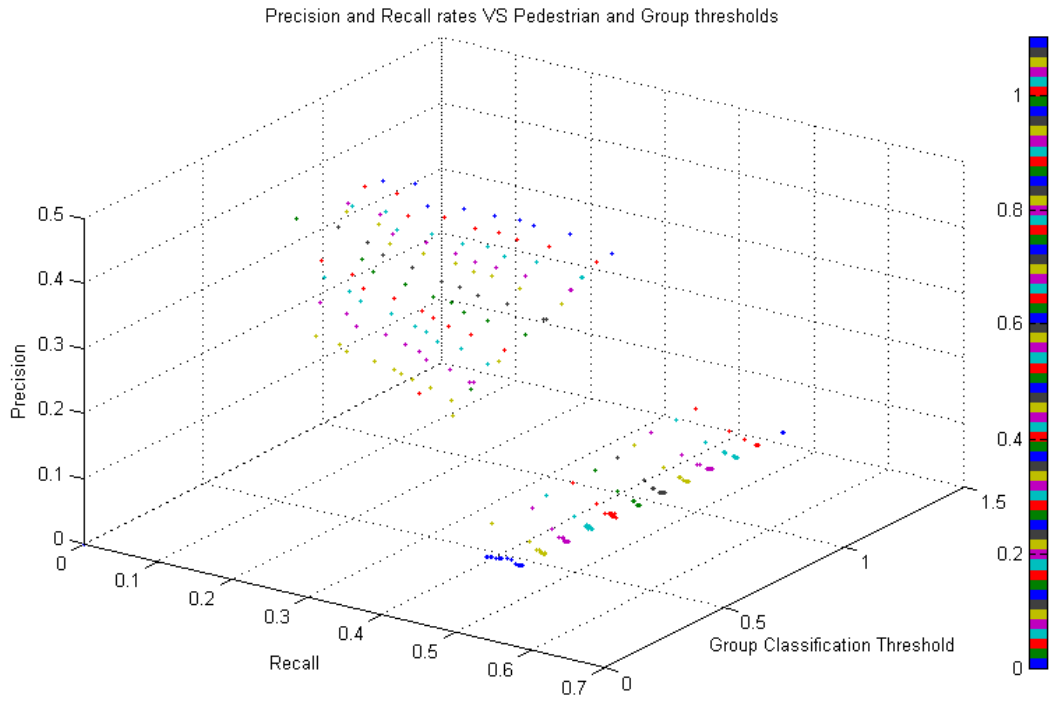


Figure 3.27: The set of *Precision VS Recall* curves obtained by modifying the two classification threshold (pedestrian, group) at the same time.

3.7.4 Quantitative results

Before presenting the results of the proposed global perception system, we detail in the two next subsections the benefit of the two main contributions proposed in this chapter: the ladar based pedestrian classification and the ability of the perception system to classify groups of people.

A data set containing $\simeq 90$ pedestrians

All the results shown in this section have been obtained by testing the algorithms detailed in this chapter on real data recorded in Paris suburb areas. A sequence of interesting situations has then been used to produce the curves presented below. This sequence is about 5 minutes long and contains approximatively 90 different pedestrians in all sorts of situations. Some views of this sequence are shown in figure 3.33.

Evaluation of the ladar based classification method

In this chapter we proposed an original and advanced methods to extract classification information from the dimensions of detected objects outlining rectangles. To assess the benefit of this approach a classification approach based only on the visible width of the objects has also been implemented and tested within the same global framework (without modifying anything

else). The object visible width is computed through the diameter of the outlining circle as proposed in some ladar based pedestrian classification algorithms (Fayad & Cherfaoui, 2007).

For simplicity, in this comparison the vision based sub-system is disabled and the classification results are directly analysed at the output of the ladar based sub-system. Besides, the group classification is also disabled $\alpha_{\text{group}} = 1$ so only $\alpha_{\text{pedestrian}}$ is used to compute *Precision VS Recall* curves.

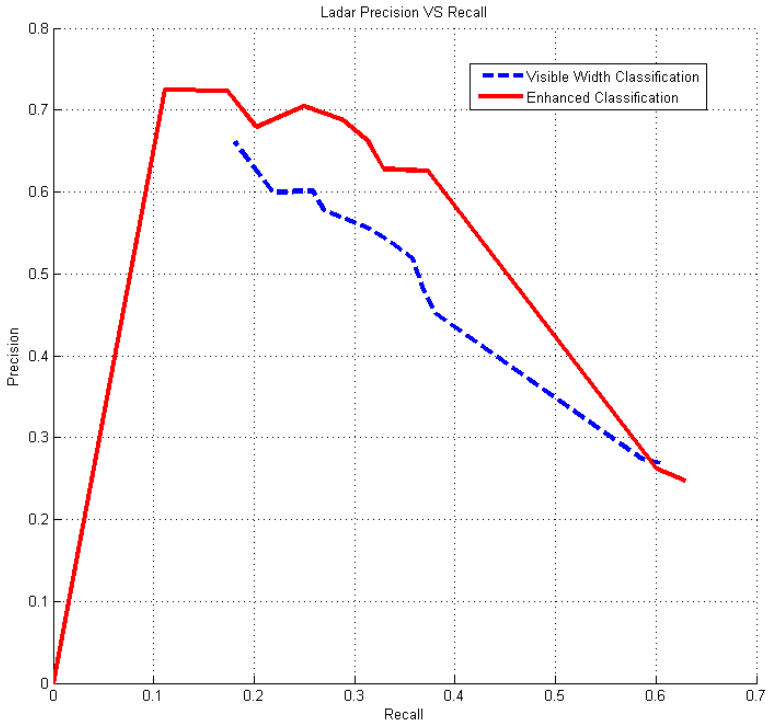


Figure 3.28: Comparative Performances of the proposed system with the proposed classification method (plain line) and the usually used visible width classification method (dashed line). Curves obtained with a varying threshold on $\Phi_{k,\text{ladar}}^{i,\text{pedestrian}}$.

The two *Precision VS Results* curves of figure 3.28 show that the proposed classification method increases both the precision and the recall rates for any chosen classification threshold.

Evaluation of the group perception capabilities

The evaluation of the system ability to classify groups of pedestrian is performed by comparing the classification performances of the ladar sub-system with and without the group classification function enabled. It is important to note that a correctly detected group of pedestrian has the ability to "valid" at the same time several pedestrians labelled as "real pedestrians" in the ground truth sequence. To counterbalance this effect, when a group of

pedestrian is wrongly detected, this is counted as a number of *n false positives* equivalent to the number of pedestrians that this group could have validated.

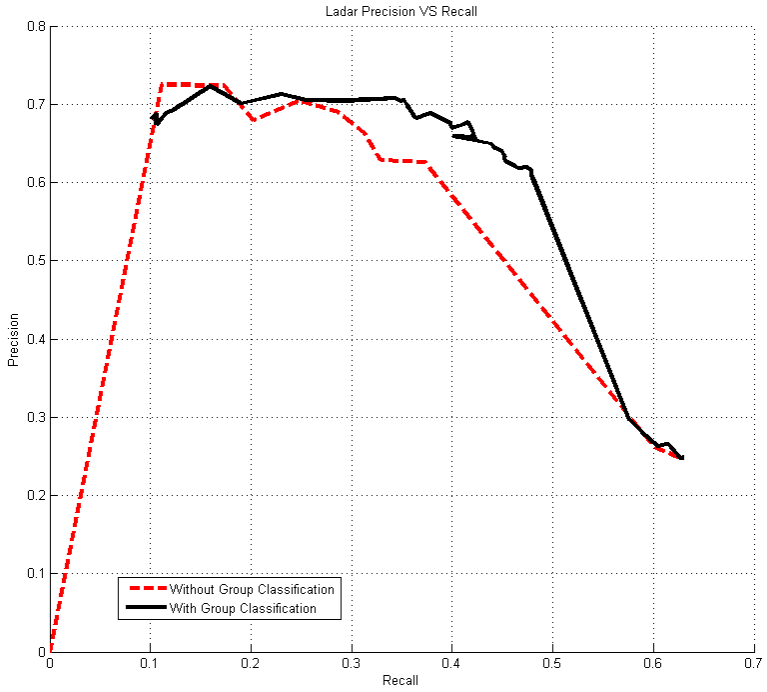


Figure 3.29: Comparative performances of the proposed system with Group classification enabled (plain line) and disabled (dashed line).

As shown in figure 3.29, the ability of the system to "see" groups of people allows a benefit on both the precision and recall rate of the system. The performances of this sub-system are illustrated in figures 3.32.

Evaluation of the sensors combination strategy

Finally it is interesting to compare the benefit of the sensors combination strategy presented in this chapter over a single sensor solution. The vision based sub-system has not been designed here to be used alone. On the contrary, the ladar based sub-system can give satisfactory results when used alone. A first comparison of the global system and on the ladar sub-system performances is given in figure 3.30.

As expected the precision of the global system is significantly enhanced compared to the precision of the ladar based sub-system. However, this precision is obtained to the detriment of the recall rate. It can be observed in practice that if lots of erroneous pedestrian candidates are efficiently filtered out by the vision based sub-system, good pedestrian candidates can also be regularly wrongly screened. This results in a "blinking effect" that impacts directly

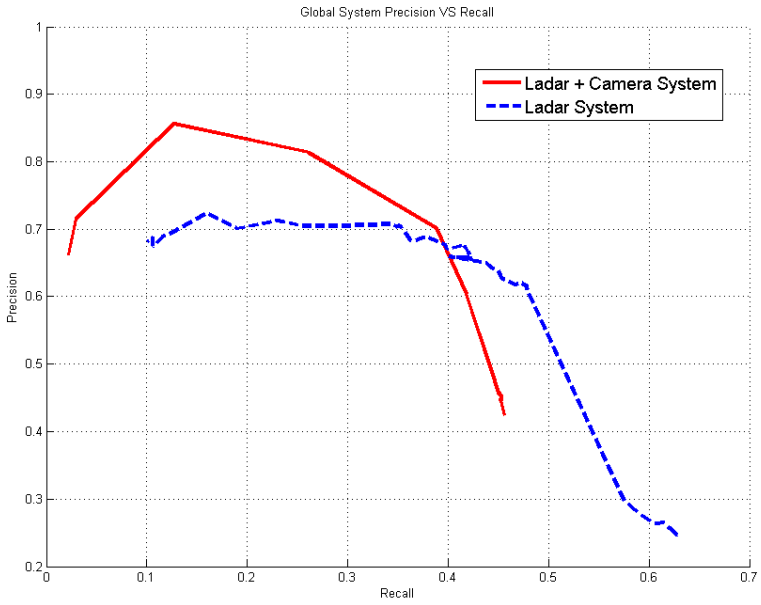


Figure 3.30: Comparative performances of the fusion based global system proposed in this chapter (plain line) over the ladar only based sub-system (dashed line).

the global system recall rate. This blinking "effect" is mainly due to the hard thresholding that is performed at the end of the process for definitive classification.

By using an additional Kalman filter based tracking algorithm after thresholding, this effect can be significantly attenuated. The benefit of the sensing strategy becomes then more homogeneous as both precision and recall capabilities are now enhanced as seen in figure 3.31. Typical outputs of the proposed system are shown in figures 3.33 to illustrate its capabilities in some interesting situations.

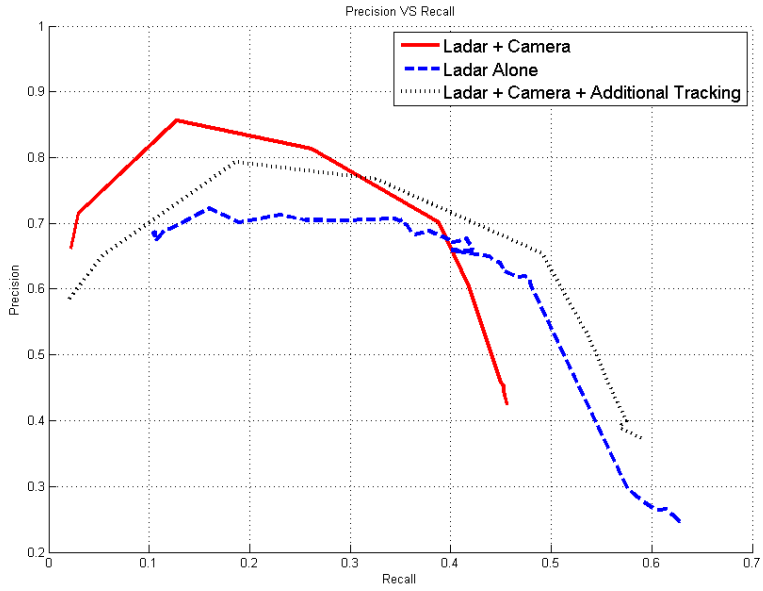


Figure 3.31: Comparative performances of the fusion based global system proposed in this chapter (plain line), the ladar only based sub-system (dashed line) and the global fusion based system where an additional simple tracking algorithm is used after thresholding (dotted line).

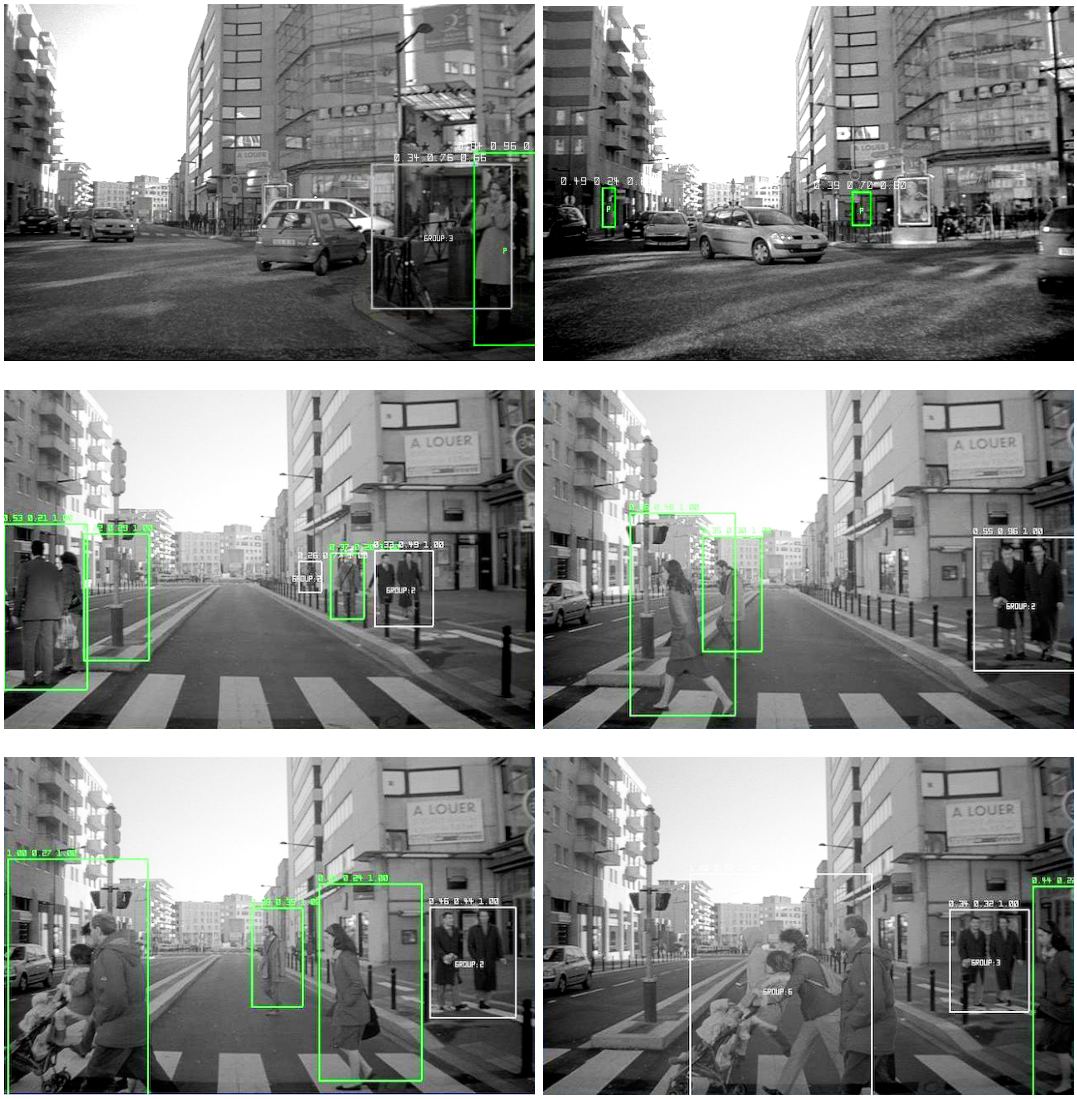


Figure 3.32: Example of the proposed ladar sub-system output in real situations. Obstacles detected as groups are shown as white rectangle. The three numbers visible above each rectangles are the three estimated scores mentioned in this chapter. Note that three first figures show obstacles being wrongly classified as groups or pedestrians.

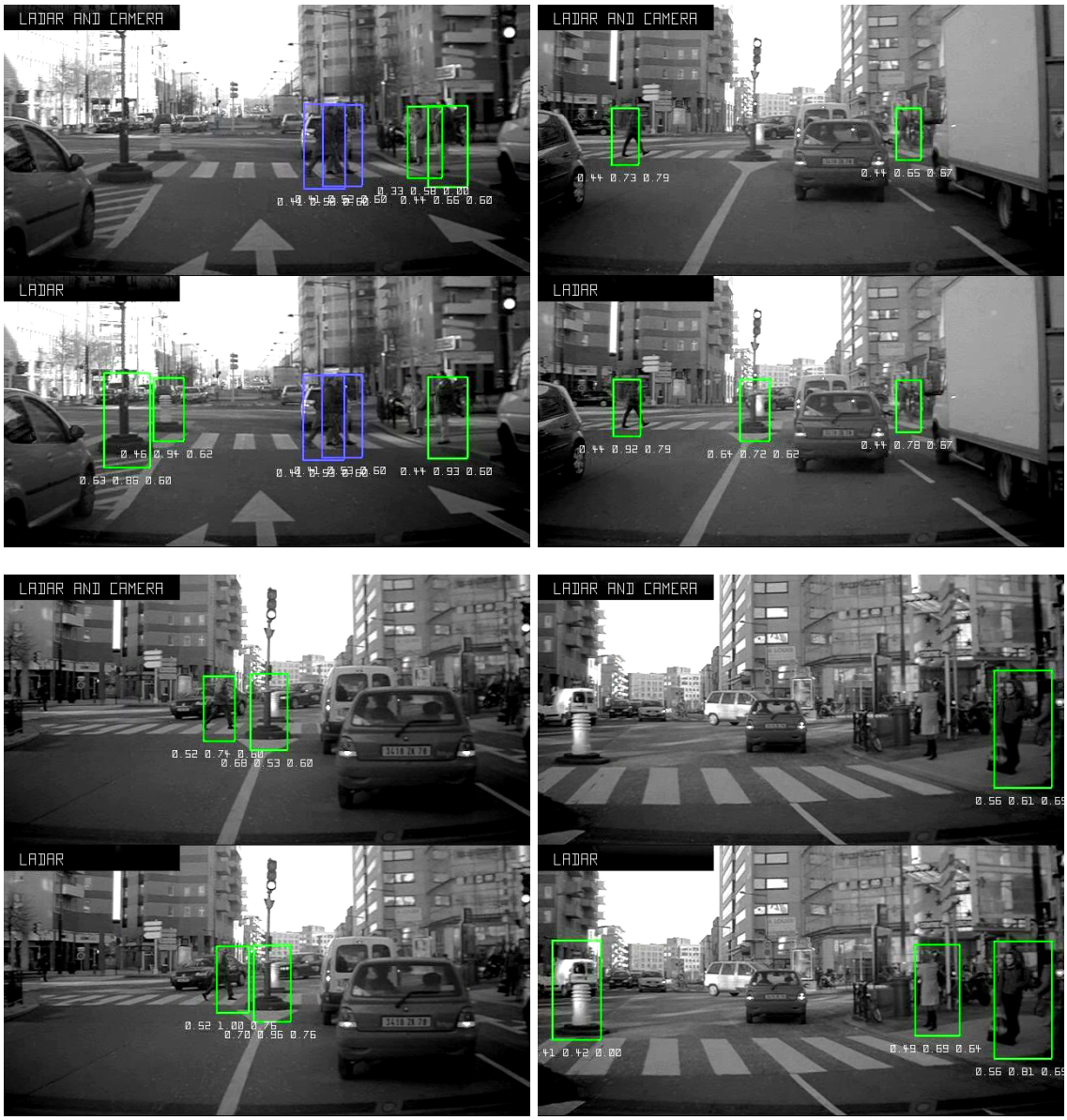


Figure 3.33: Example of the proposed complete system output in real situations. The corresponding output of the ladar sub-system is also given for comparison below each snapshot. Obstacles detected as groups are shown as blue rectangles (the basic routine mentioned in section 3.5.2 is used here to discriminate roughly pedestrians inside "groups"). The three numbers visible below each rectangles are the three uncertainty scores mentioned in this chapter. The two last figures show situations where the system fails to compute the correct list of pedestrians present in the scene.

3.8 Conclusion

In this chapter a complete pedestrian perception system has been described. This system intended to be integrated to an onboard precrash setup has been designed to achieve a high precision rate (85 %) and to be easily implemented on onboard *CPU* units. Relevant trade-offs between efficiency and complexity have systematically been found to propose a realistic solution to pedestrian perception in highly changing environments. When executed on a desktop single 2GHz core, the global system takes $11ms$ on average to process an entire ladar scan and provide a new list of pedestrians and groups.

While quite satisfactory in lots of situations, this perception system is of course not flawless. Some pedestrians will indeed stay inexorably undetected, while some specific obstacle configurations will unfortunately always produce some erroneous detections. If some of these limitations can undoubtedly be overcome by using more efficient detection, classification or tracking algorithms, we believe that most of them can only be overcome by using totally different approaches.

In the following chapter, a complete analysis of the fundamental limitations of existing *DETAC* systems (such as the one presented in this chapter) is given and a novel approach to *DETAC* problems is then proposed in Chapter 5 and 6.

Résumé en français du chapitre 4

Nous nous appuyons, dans ce chapitre, sur le système de perception proposé au chapitre 3 pour fournir une étude détaillée des 4 difficultés principales qu'aucun système de perception actuel n'est capable de contourner simultanément et qui sont, à notre avis, les causes de leur manque de fiabilité. Nous appuyons cette analyse sur des exemples de situations concrètes où le système présenté au chapitre 3, comme tout autre système équivalent, échoue à fournir une analyse satisfaisante de la scène. Les performances des algorithmes dit "SLAMMOT" récemment proposés dans la litérature sont aussi discutées.

Chapter 4

Fundamental Limitations of Existing Approaches

Contents

4.1 Introduction

As seen in the previous chapter, a perception system based on common *DETAC* approaches is able to achieve a satisfactory level of precision in lots of urban situations. However, this level of performances is still arguably far under what is required for efficient precrash setups. As such perception strategies are indeed intended to interfere with the driver, a far better precision and recall rates are likely to be required. In fact, if most common situations can perfectly be handled by common *DETAC* systems, such systems will inexorably fail in some specific but yet common situations.

These failures are particularly problematic as they cannot be easily overcome using better detection, tracking or classification algorithms. They are instead due to the fundamental limitations imposed by the common sequential approaches used to solve *DETAC* problems.

In this chapter, a complete description of these critical limitations is given and because some efforts have recently been made to address some of them, an overview of the solutions that already exist for each problem is also presented. However, these recent approaches are still not capable to solve simultaneously all the limitations presented in this chapter. That is why, we will propose in the next chapters an original solution that addresses simultaneously all these needs.

4.2 Detection failures

4.2.1 The geometry based detection problem

The main issue with most *DETAC* systems is arguably the fact that detection is performed first and independently from tracking. For ladar based systems, this leads to detection algorithms being only based on some geometrical criteria as no other information (such as velocities) is directly available.

Unfortunately, in dense environments, objects are regularly located at small distances from each others and discriminating each of them can be hard using only geometrical features. This is for example the case when pedestrians are walking along other bigger obstacles (buildings, vehicles, etc...). In figure 4.1, the close proximity of the pedestrian and the vehicle prevents the pedestrian from being correctly discriminated by the system presented in chapter 3.
It is important to note that decreasing the threshold d used in the detection algorithm is not satisfactory either. If the pedestrian is eventually correctly detected, many bigger obstacles will then probably be detected as several small objects leading to other issues.

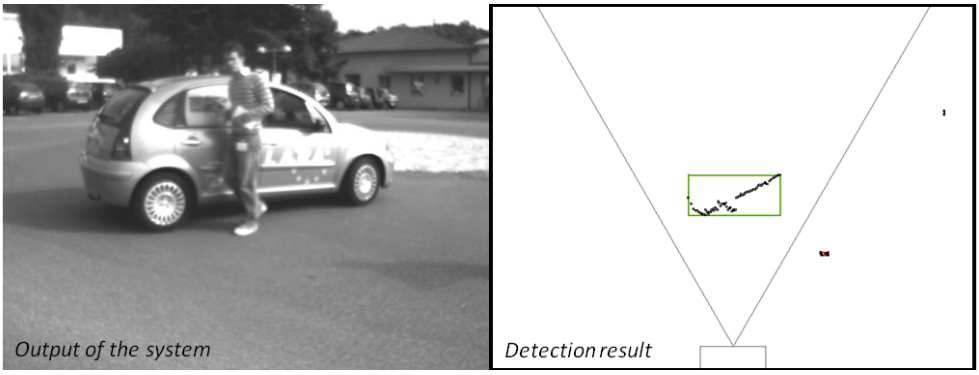


Figure 4.1: Example of a problematic detection failure.

The same detection problem also occurs on pedestrian crossings where people are close from each others but also heavily occluded from sensor observations. Figure 4.2 shows a situation where several pedestrians are crossing the street, the detection performed by the system presented in chapter 3 is also shown. As expected, the detection algorithms are unable to handle the situation and the output released by the system are incorrect.

4.2.2 Solutions

Two solutions can be proposed. Modifying the sensing strategy of the system is the first one. Some radars are collecting velocity measures about obstacles and could presumably handle such situations. However, the geometrical information collected about each object is likely to be less precise than with ladar. This is unfortunately likely to make subsequent classification or tracking problems harder.
Several ladars can also be used to prevent occlusions as proposed in (Zhao & Shibasaki, 2005) where several static ladars are used to detect pedestrians. Unfortunately, this solution is not

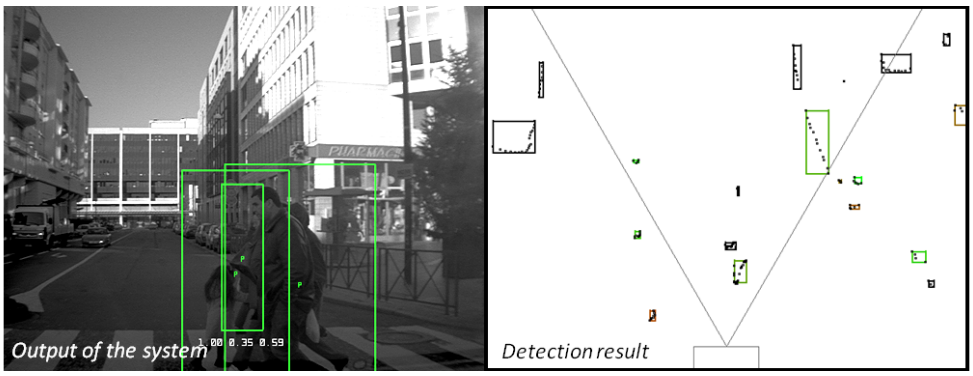


Figure 4.2: Pedestrians crossing the streets are very difficult to detect using a ladar based system.

easily scalable to mobile platforms.

A more natural solution is to infer obstacles dynamic features before detection. This is usually what the tracking algorithm is expected to do, but existing tracking approaches cannot be performed on ladar impacts directly. Indeed, a given laser ray never impacts the same physical point of an object. As a result, even if solving the data association problem was easily feasible, such a tracking approach would only track the variation of impacts ranges but not the objects of the scene. Such an impact based tracking would of course be useless here.

Tracking and detection should instead be addressed simultaneously or at least concurrently. A very interesting approach to that problem is proposed in (Vu & Aycard, 2009) following the work of (Petrovskaya & Thrun, 2008). The authors use a sampling approach (*Monte-Carlo Markov Chain*) to solve simultaneously the detection and tracking problems for moving objects. Although, the proposed method is only applicable to objects whose outlines can be correctly approximated with basic primitives (squares, lines, points), preliminary results are encouraging. However, in the situation shown in figure 4.2, it is not clear wether crossing pedestrians can correctly be matched with basic primitives. The insightful solution proposed in (Vu & Aycard, 2009) should be further tested in specific difficult configurations to evaluate its benefit in this case.

It is clear from this analysis that this detection problem can only be address by enabling a higher level of interaction between the Detection and the Tracking tasks.

Problem	Interaction needed
Geometry based Detection	Detection $\Longleftrightarrow$ Tracking

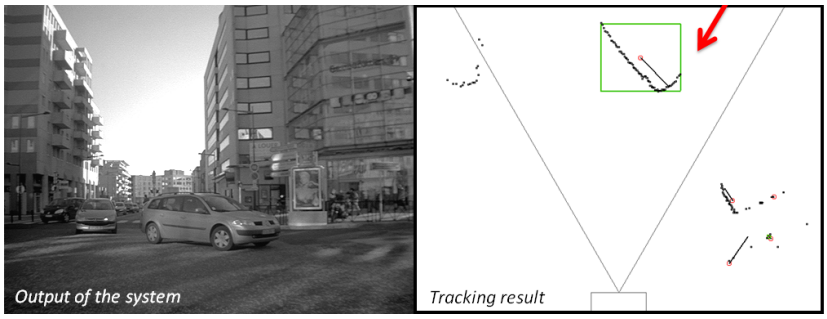
4.3 Tracking failures

4.3.1 The point-based tracking problem

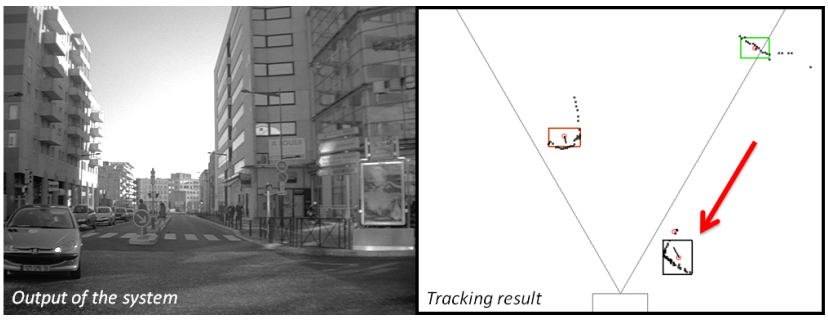
Most existing tracking approaches are based on an object point-based representation. Sadly, tracking the same object point over time is not an easy task. An object point visible at time k can indeed be occluded at time $k+1$.

For this reason, it is usually easier to track a "virtual" point of an object (eg. its center of gravity) than one of its physical point. Even if the orientation of the object changes, it is then still possible to compute a good estimate of its center of gravity. Unfortunately this approach becomes problematic when tracked objects are heavily occluded. In such situations, the estimated position of the object center of gravity is likely to be erroneous leading directly to non relevant velocity estimates or critical data association failures.

In figure 4.3 a typical situation where a vehicle is improperly tracked due to that problem is shown. In this case, the only consequence of the tracking failure is an incorrect velocity estimate for the considered vehicle.



(a) Time $t = t_0$

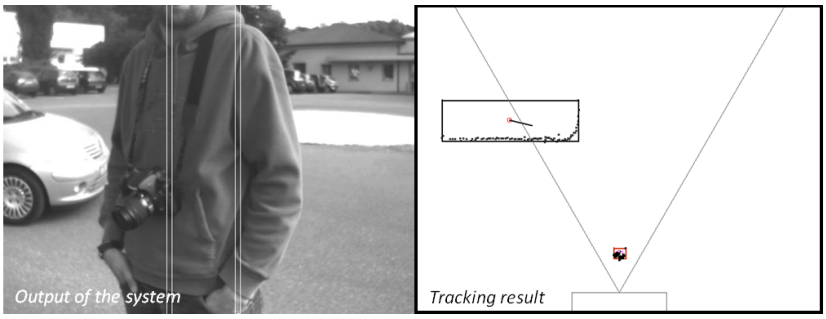


(b) Time $t = t_0 + T$

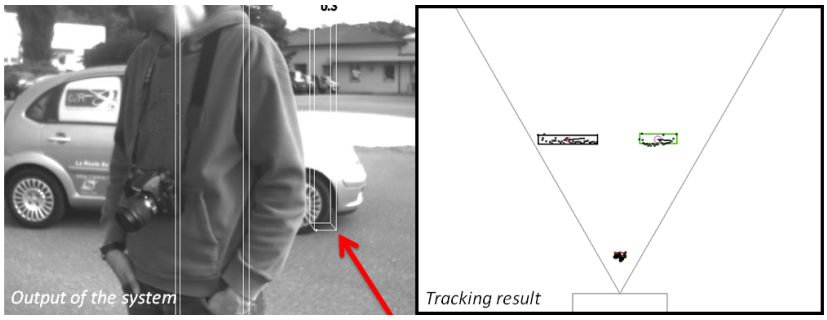
Figure 4.3: Situation where the point based tracking is not sufficient to track a moving vehicle (indicated by the red arrow).

Furthermore, this tracking limitation can also produce data association errors that are likely to generate misclassified system outputs as shown in figure 4.4. In this specific situation, a pedestrian located in front of the sensor is occluding the middle portion of a passing vehicle.

The *JPDA* based tracking algorithm is unable to track correctly this vehicle and generates a false pedestrian detection. The same result would also probably be obtained with any other point-based tracking scheme.



(a) Time $t = t_0$



(b) Time $t = t_0 + T$

Figure 4.4: Situation where a data association failure leads to a *false* pedestrian generated by the system and indicated on the image by the 3D white parallelepiped (red arrow).

4.3.2 Solutions

These tracking failures due to point-based approaches can naturally be solved by tracking objects real outlines. As mentioned above, this is not an easy task as the visible outlines are expected to vary a lot over time.

(Wang *et al.*, 2007) propose to store the current visible outlines of every tracked object in an occupancy grid. Tracked objects and new detected objects are then associated using first a *multi-hypothesis (MHT)* algorithm for rough association and then an *Iterative Closest Point (ICP)* based algorithm to refine the registration. This approach is an interesting solution to the considered problem but is still highly dependent on accurate detection.

Interestingly, the simultaneous detection and tracking approach proposed in (Vu & Aycard, 2009) and mentioned in the previous section allows also to solve the tracking failures related to point-based representations. By approximating objects physical outlines by a parametric shapes (rectangles, lines or points), the proposed method is indeed capable of handling correctly the situations presented above.

It is important to note that the problem of tracking physical outlines is thus strongly related to the way tracked objects outlines are approximated and stored. In that sense, this tracking problem can be seen as closely related to the problem of moving object mapping that is discussed in the next section.

Problem	Interaction needed
Point-based Tracking	Tracking $\Longleftrightarrow$ Moving object Mapping

4.4 Classification failures

4.4.1 The "*stair effect*" classification problem

Classification failures can obviously be induced by previous detection or tracking problems. But there are also specific situations where classification can fail even if tracking and detection are performed correctly. We are not referring here to situations where the classification task is made very difficult by too few relevant sensing data for example. In that case, even a perfect perception system would fail. We are interested instead in situations that lead to classification failures even if the data collected by the sensor over time is still informative enough to allow any trained person to identify easily the nature of any objects in the scene.

This type of classification failures regularly occurs when the same real object appear under various outlines in a short sequence of raw data. A caricatural example is shown in figure 4.5. Because of the sensing platform pitching, the stair is never entirely visible on one single scan (single layer ladar), but is easily identifiable by looking at few scans.

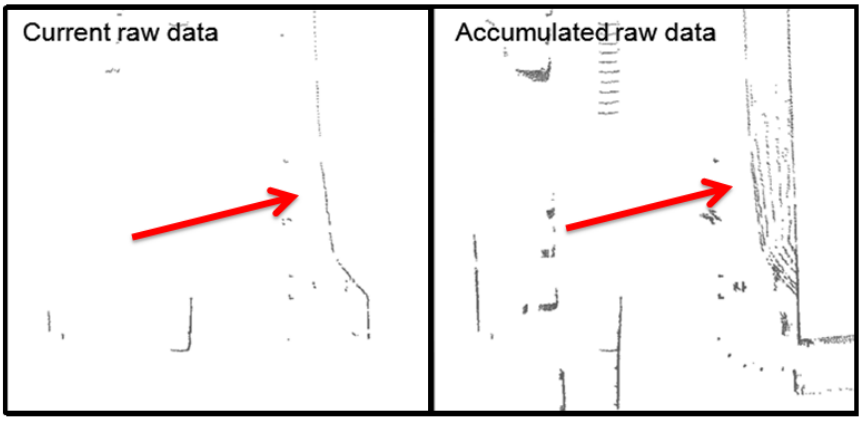


Figure 4.5: Illustration of the mapping benefit for classification

This "*stair effect*" can in fact happen with all objects (static or moving) having different shapes when horizontally cut at various height (cars, trucks, trees, etc...). In real situations, the ladar layer is likely to hit the same object at different heights over time and to produce impact configurations that are not easily identifiable by looking at just one scan.

A classification failure example due to that problem is shown in figure 4.6. A truck is indeed seen by the ladar as an unordered set of points that is really difficult to handle on any single scan. As a result, the end of the truck is here wrongly classified as a group of people by the ladar based sub-system proposed in chapter 3. With single layer ladars, we believe that such obstacles can only be correctly classified by aggregating overtime relevant information about their physical outlines.

Note that using multi-layer sensing technologies in a appropriate way might mitigate this problem. However, in the situation shown in figure 4.6, all of the four ladar layers are hitting the frame of the truck and collect spurious data.

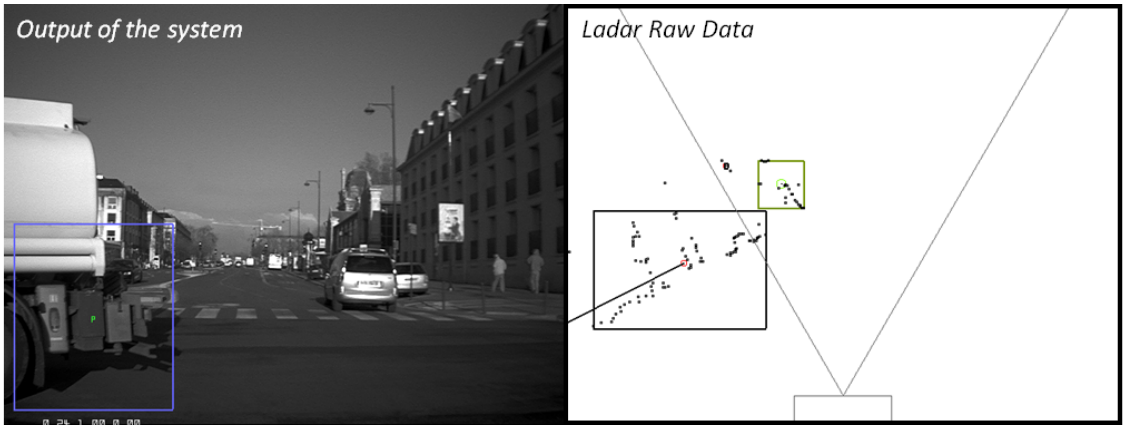


Figure 4.6: An example of a moving obstacle that cannot be easily identified on a single ladar scan. The system presented in chapter 3 wrongly classifies the rear of the truck as a group of people (blue rectangle on the camera image).

4.4.2 Solutions

Aggregating the information over time to construct objects real outlines is what the mapping task is expected to do. Unfortunately, as described in chapter 2, most existing mapping approaches are based on a static world assumption and cannot be directly used to address these classification failures. It is important to note that most works related to dynamic environments mapping propose methods to build a consistent maps of static objects by correctly filtering the moving objects. But mapping of moving objects is usually not addressed.

This problem is in fact strongly related to the tracking problem mentioned above. If a tracking algorithm is capable of registering accurately new observations with the stored outlines of the currently tracked objects, it is indeed also capable in principle to build over time a relevant map for each tracked object.

The two interesting approaches that are able to solve the tracking problems mentioned above are thus good candidates for that problem too. Unfortunately, as handling unstructured obstacles is a prerequisite here, the method proposed in (Vu & Aycard, 2009) cannot be

adapted to cope with this mapping problem. As already mentioned, this approach is indeed based on the assumption that moving objects are structured enough to fit basic primitives.

On the contrary, the *SLAM* with *DATMO* approach proposed in (Wang *et al.*, 2007) could in principle be extended to moving object mapping, provided that the objects are initially correctly detected. Indeed, unlike (Vu & Aycard, 2009), this approach do not solve the critical detection problems described above.
From this analysis, it is clear that this classification issue called "stair effect" problem is strongly related to the problem of moving objects mapping.

Problem	Interaction needed
"*Stair effect*" Classification problem	Classification $\Longleftrightarrow$ Moving objects Mapping

4.5 Uncertainty management of interacting tasks

We saw in the previous sections of this chapter that recent *SLAM with DATMO* approaches can locally solve some of the tough problems discussed above. These approaches, based on an increased level of interaction between some of the perceptual tasks raise a set of new questions related to uncertainty management. These issues are discussed in the following paragraphs.

4.5.1 The problem of heterogeneity for interacting tasks

As stated several time before, maintaining throughout the perception process an appropriate knowledge about the uncertainties generated by the successive perception algorithms is critical. This knowledge is indeed mandatory to allow at the end of the process optimal decisions to be made by the system.

These uncertainties are however usually related to high dimensional continuous spaces and are as such always difficult to compute and to store.

Consequently, all existing perceptual algorithms handle a certain level of uncertainty through specific assumptions and using various different representations. In the same perception system, uncertainties might for example be modelled by Gaussians in tracking, occupancy maps in mapping and through *Monte-Carlo* samples in localization.

This is of course due to the fact that the uncertainty representation is directly dependent on the nature of the algorithm that is used to compute a perceptual task. Because most perceptual tasks have historically been addressed separately, a great variety of uncertainty representation have been adopted. Consequently, perception systems that make use of these algorithms are usually dealing simultaneously with different uncertainty representations.

This heterogeneity leads to theoretical and practical problems when an algorithm that uses a particular mathematical formalism for uncertainties is expected to handle the uncertainties

generated by another algorithm using a different framework. Indeed, a specific uncertainty representation (eg. sampling representation) is usually directly linked to some specific assumptions (eg. nonlinear models, multi-modal densities) and as such can not necessarily be converted to a different framework (eg. Gaussians).

4.5.2 The problem of ML and MAP estimates for interacting tasks

This second problem can be seen as a direct consequence of the heterogeneity problem. Indeed a fast but sub-optimal way to make interactions between algorithms easier is to do it through most probable estimates (*ML* or *MAP* estimates) that are deprived of their uncertainty information.

In the recent literature related to *SLAM* with *DATMO* systems, the localization result is for example usually transmitted to *DATMO* algorithms as a most probable estimate. Any localization failure can in this case presumably lead to moving objects detection and tracking failures.

In *SLAM with DATMO* approaches where the objects classification as moving or static is conversely only incorporated in the *SLAM* computation as a most probable estimate, *DATMO* failure is likely to induce *SLAM* failures in the same manner.

In other words, instead of being mutually beneficial, *SLAM* and *DATMO* can easily become mutually destructing if uncertainties are not handled with care. This situation is schematized in figure 4.7.

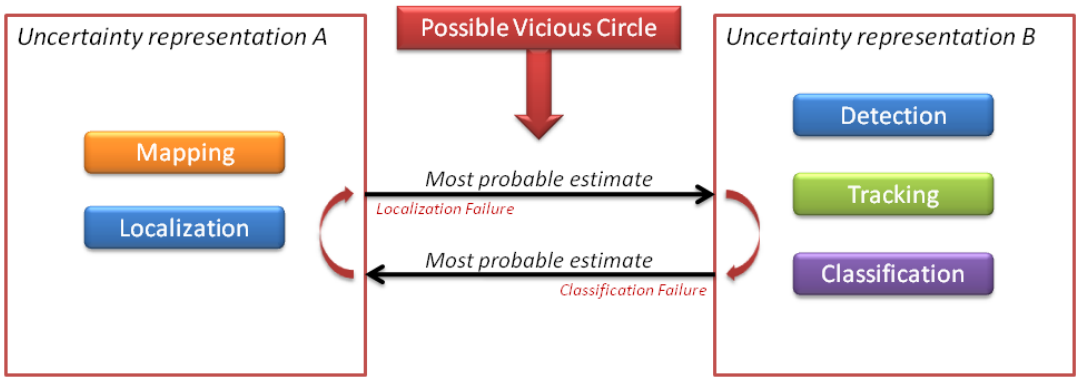


Figure 4.7: Illustration of a possible vicious circle that can be initiated if a localization or a classification failure occurs.

4.5.3 Summary

As detailed in this chapter and as proved by recent works, reliable perception systems can be built by addressing concurrently the five perceptual tasks. But we believe that the benefit of exploiting many interactions between the tasks can only be ensured:

1. if a common mathematical framework is used to compute and to store algorithms uncertainties related to all the interacting tasks.
2. if most probable estimates are not used in such a way that a vicious circle can be initiated.

An schematic example of an interacting perception system that meets these two criteria is given in figure 4.8. When an algorithm receive an information as a full posterior density (*Bayesian* estimate) from another task, a failure vicious circle is less likely to happen. Indeed, even if the most probable estimate of this full posterior density is erroneous, this density contains also the true estimate. Providing that this true estimate has still a significant weight in the density, the receiving algorithm has a chance to recover from the failure of the sending algorithm.

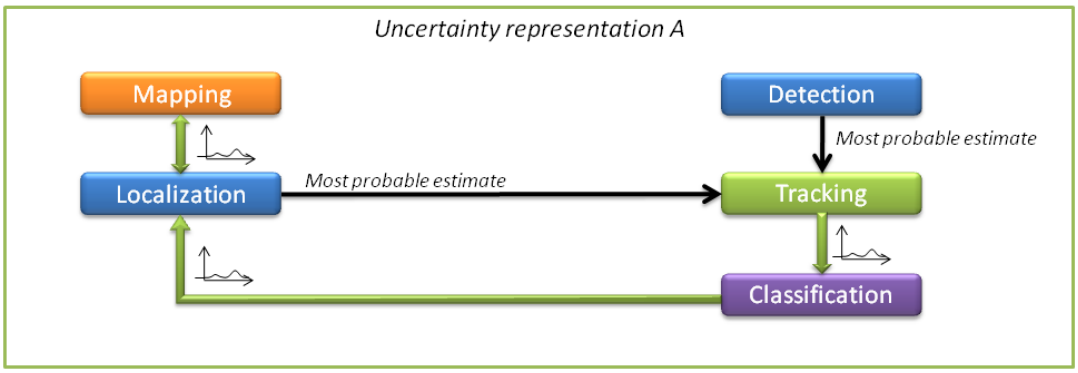


Figure 4.8: Example of an interacting perception system with a satisfactory uncertainty management.

4.6 Conclusion

In this chapter a detailed description of the main limitations undergone by current *DETAC* systems is given. This analysis is based both on the results of the pedestrian perception system proposed in Chapter 3 and on *SLAM with DATMO* approaches recently proposed in the literature.

Four main problems have been identified:

1. The geometric based Detection problem
2. The point-based Tracking problem
3. The "*stair effect*" Classification problem
4. The problem of uncertainty management for interacting tasks

If no existing approaches is capable of solving all this problems at the same time, recent works in the *SLAM with DATMO* literature are locally capable of solving on or several of these issues. It is interesting to observe that the three first issues can be solved by pushing the level of interaction between the perceptual tasks to a higher level.

Problems	Interaction needed
Geometric-based Detection	Detection $\Longleftrightarrow$ Tracking
Point-based Tracking	Tracking $\Longleftrightarrow$ Mapping
"*Stair effect*" Classification	Classification $\Longleftrightarrow$ Mapping

The fourth problem, related to uncertainty management, can be solved by using a common mathematical formalism to address interacting tasks and by avoiding cycling use of most probable estimates.

This analysis motivated the work presented in the next chapters where an original grid-based approach is proposed to solve simultaneously these four problems.

Résumé en français du chapitre 5

Ce chapitre pose les bases théoriques d'une approche basée sur une représentation par grille du problème de perception qui permet de contourner simultanément les 4 écueils présentés au chapitre 4. Un présentation générale des principales étapes d'un algorithme de perception reposant sur cette approche est aussi donnée.

Chapter 5

Grid-based Global Approach for Reliable Perception

Contents

5.1 Introduction

In Chapter 4, four important limitations that affect most existing perception systems were identified. It has been shown that while three of them can be efficiently overcome by using appropriate interactions between perceptual tasks, a careful uncertainty management strategy needs also to be deployed to ensure mutually beneficial interactions.

To our knowledge, none of the existing approaches to perception problems can be extended to cope with these four issues at the same time. The reason for this is simple: handling these issues is computationally demanding and most existing approaches are designed to be usable on reasonable onboard architectures.

The computational criteria should however be considered with care. If it is true that, with limited processing units, being fast will forever be a fundamental requirement for perception systems, this should not prevent more demanding approach from being investigated.

Following this idea, we propose in this chapter to compute the five perceptual tasks using a unified mathematical formalism. This framework based on grids of cells allows first to naturally compute discrete Bayesian estimates for each task and second to incorporate elegantly all the possible tasks interactions. This chapter is only intended to present the theoretical

principles of this approach and to discuss its benefits. In chapter 6 of this thesis, a specific implementation of these concepts for pedestrian perception is described and preliminary results in real situations are presented.

5.2 Principles

5.2.1 Grid-based Uncertainty Representation

In a Bayesian framework, the notion of uncertainties can be embedded in probability density functions. Probability density functions are neither easy to compute nor easy to store. In most problems, these densities are indeed related to highly dimensional continuous spaces. To allow Bayesian estimates (*full posterior densities*) to be computed, approximations are usually needed. Mixture of Gaussians or sampling representations can thus be adapted to describe these uncertainties. However, maintaining a good density representation through samples or through Gaussians is usually not an easy task as discussed for example in (Leal, 2003).

On the contrary, grids of cells offer a polyvalent and powerful solution to that problem. Grid representations are based on a appropriate discrete approximation of a particular space. In (Elfes, 1989b), such a representation was proposed to represent uncertainties generated by mapping algorithms. Every cell was in this case representing the probability of a specific area in the environment to be occupied by an object or not.

Grids of cells can however be used to represent a great variety of discrete probability mass functions (*pmf*). This representation has two main advantages:

1. Free form multi-modal densities can be represented
2. Cells-based representation are easy to handle with a computer

But the main inconvenient of this representation is certainly the relatively high computational and memory requirements that it requires, at least in its "simple" form. Grid-based representation are indeed originally based on a uniform repartition of the cells over all the available space. This can be problematic in practice as the number of cell required to reach a good representation of large spaces grows very quickly. This problem has nevertheless received over the last decade some interesting solutions such as those discussed in (Kraetzschmar *et al.*, 2004; Montemerlo & Thrun, 2004).

Our approach is based on the idea that grids of cells can be used as a unified mathematical formalism to

1. Compute naturally every perceptual task.
2. Store efficiently the results of these computations as probability mass functions.
3. Allow any given perceptual task to elegantly interact with any other task.

In this chapter, the probability mass functions (noted *pmf*) that should be computed to solve each perceptual task are first detailed. Then, two mathematical methods are discussed to simplify the intractable joint probability mass functions that arise in the computations. Finally, the outlines of a global algorithm solving sequentially every perceptual task in this framework are described.

5.2.2 Perceptual Tasks description in a Grid-based Representation

The Mapping problem in a Grid-based Representation

The area of interest around the sensing platform is represented as a grid of N cells noted $E = \{x_i\}_{1 \leq i \leq N}$, where x_i refers to the centroid of a square cell of dimension a. Let M_k be a random variable such that at time k, $\forall x_i \in E$,

$$M_k(x_i) = \{ \begin{array}{l} 1 \text{ if } x_i \text{ is occupied} \\ 0 \text{ if } x_i \text{ is not occupied} \end{array}$$

The mapping problem at time k can then be expressed as the estimation of the following probability mass function *(pmf)*:

$$\boxed{P(M_k(x_i) = 1 | Z_{0:k}) \quad \forall x_i \in E} \tag{5.1}$$

This *pmf* can be represented as a usual occupancy grid as shown in figure 5.1.

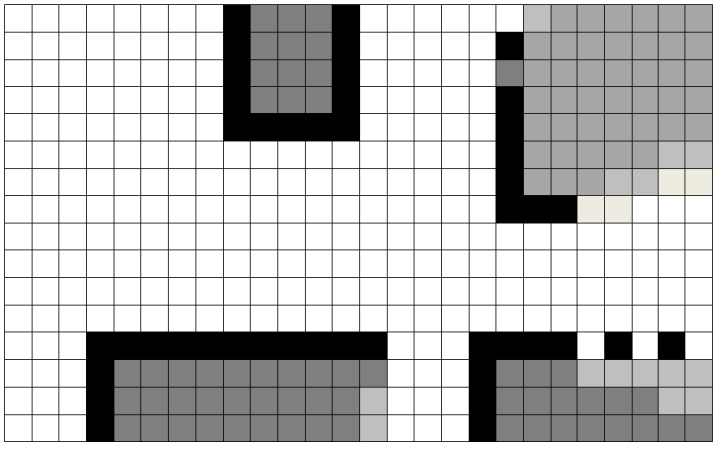


Figure 5.1: A schematic view of the Mapping *pmf*.

The Localisation problem in a Grid-based Representation

Let L_k be the discrete random vector that represent the state vector of the sensing platform at time k:

$$L_k = \begin{pmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{pmatrix}$$

The localisation problem at time k can then be expressed as:

$$\boxed{P(L_k = l_j | Z_{o:k}) \qquad \forall l_j \in E \times V} \tag{5.2}$$

, where V is the discrete space of possible speeds. Because the dimensionality (4D) of this probability mass function is too high to be correctly represented a schematic view of this *pmf* is given in figure 5.2.

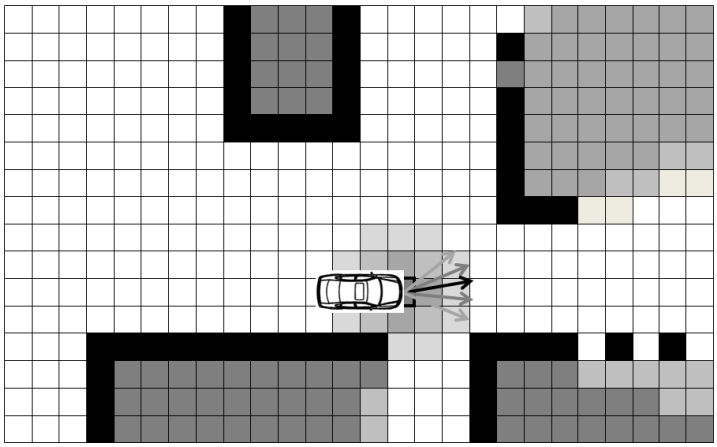


Figure 5.2: A schematic view of the Localization *pmf* superimposed with the Mapping *pmf* mentioned above.

The Tracking problem in a Grid-based Representation

As stated in chapter 2, the tracking problem has two objectives:

1. Find the correspondence between the new incoming data and the already tracked entities. This is referred to as a *registration* or *(data-)association* problem.

2. Use the association results to make estimations about the dynamic features of these entities (velocity, acceleration, etc...). This can be seen as a *filtering* problem. However, because many other estimations in this chapter can be seen as *filtering* problems, this particular estimation will be referred to as the *Velocity Estimation* problem for clarity.

Association subproblem

By noting $X_{k-1}^{next}(x_i)$ the coordinates at time k of the physical point located in x_i at time $k-1$, the *Association* problem can be seen as the estimation of the following *pmf*:

$$\boxed{P(X_{k-1}^{next}(x_i) = x_j | M_{k-1}(x_i) = 1, Z_{0:k}) \qquad \forall (x_i, x_j) \in E^2} \tag{5.3}$$

Velocity Estimation subproblem

We note $V_k(x_i)$ the random variable that represents the dynamic features of the physical point that occupies the cell x_i at time k (such as its velocities, accelerations, etc...).

As this problem is only relevant if the corresponding cell is *occupied*, the *Velocity Estimation* problem can be regarded as the estimation of the following *pmf*:

$$\boxed{P(V_k(x_i) = v_j | M_k(x_i) = 1, Z_{0:k}) \qquad \forall (v_j, x_i) \in V \times E} \tag{5.4}$$

It is important to note that the above formulation implies that the Tracking will be performed at the cell level. This will be discussed later in this chapter.

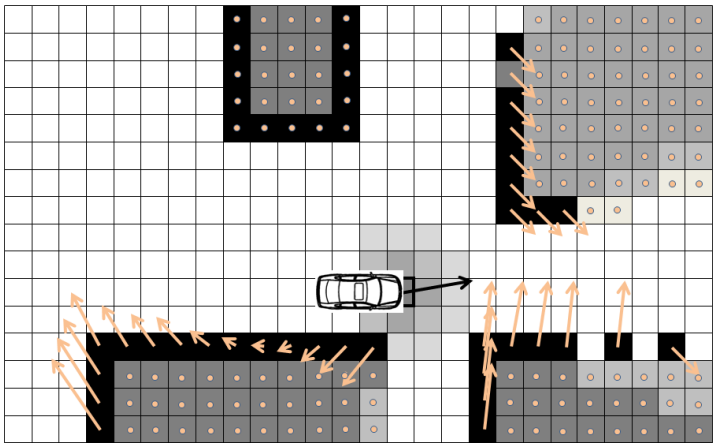


Figure 5.3: Schematic view of the Velocity Estimation *pmf* (5.4) superimposed with the other *pmfs* described above.

The Detection problem in a Grid-based Representation

Let us note $D_k(x_i, x_j)$ the binary random variable that is only equal to one if the cells x_i and x_j are occupied by the same object. The detection problem can then be rewritten as the

estimation of the following *pmf*:

$$\boxed{P(D_k(x_i, x_j) = 1 | M_k(x_i) = 1, M_k(x_j) = 1, Z_{0:k}) \quad \forall (x_i, x_j) \in E^2} \tag{5.5}$$

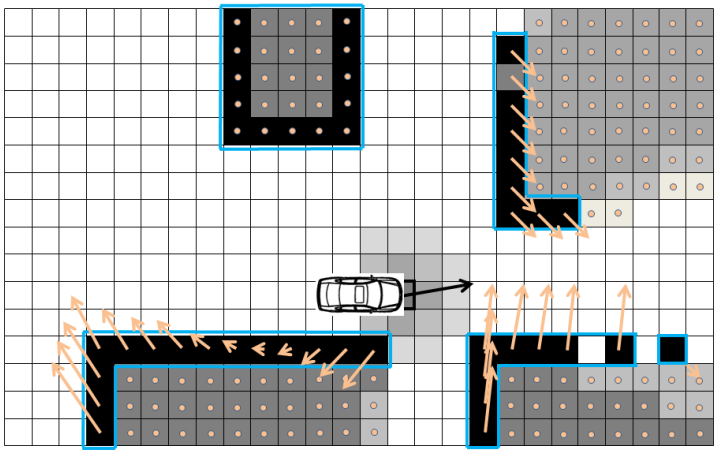


Figure 5.4: Schematic view of the Detection *pmf* (represented here as most probable estimates for simplicity) superimposed with the other *pmfs* described previously.

The Classification problem in a Grid-based Representation

We define C as the set of all possible *types* of objects present in the scene and $C_k(x_i)$ the random variable that refers to the class to which the object occupying the cell x_i belongs. The classification problem can be seen as the estimation of the following *pmf*:

$$\boxed{P(C_k(x_i) = c_j | M_k(x_i) = 1, Z_{0:k}) \quad \forall (c_j, x_i) \in C \times E} \tag{5.6}$$

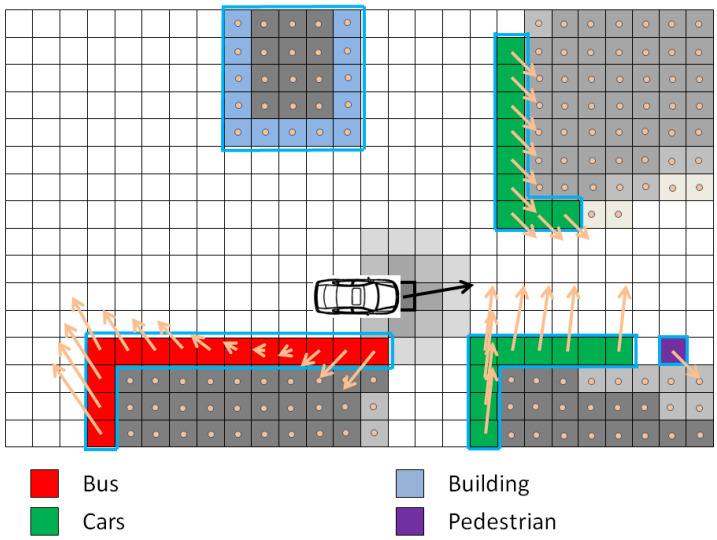


Figure 5.5: Schematic view of the Classification *pmf* (represented here as most probable estimates for simplicity) superimposed with the other *pmf*.

5.3 Joint Probability Mass Functions Approximations

One of the objectives of this chapter is to propose an elegant way to compute at each cycle and for all the cells $x_i \in E$, the 6 probability mass functions described in the previous sections. Unfortunately, each one of these computations will induce a marginalization (weighted summation) over highly dimensional joint probability mass functions that are intractable in most cases.

Indeed, in the proposed grid-based representation, a given local random variable $A_k(x_i)$ (with A referring to any random variable introduced above) is usually strongly dependent on the value taken by the random variable on neighbouring cells $A_k(x_j)$ ($\forall x_j \in E$), as some of the neighbouring cells are likely to be occupied by the same real object. Hence, the computation of $P(A_k(x_i)|Z_{0:k})$ should then be derived from the above joint *pmf*:

$$P(A_k(x_i)|Z_{0:k}) = \sum P(A_k(x_1), ..., A_k(x_N)|Z_{0:k}) \tag{5.7}$$

$$= \sum P(A_k|Z_{0:k}) \tag{5.8}$$

, where variables of the form A_k refer to the random vector $(A_k(x_1), ..., A_k(x_N))$.

Besides, because all perception tasks are interdependent problems, a given local random variable $A_k(x_i)$ is also dependent on the random variables related to the other perceptual tasks at any other location $(B_k(x_j), C_k(x_l), ...)$ $\forall x_j, x_l, ... \in E$.

Consequently, the computation of any given *pmf* related to a specific perceptual task needs in principle to be derived from the following general joint *pmf*:

$$P(A_k(x_i)|Z_{0:k}) = \sum P(M_k, L_k, V_k, D_k, C_k, X_{k-1}^{next}|Z_{0:k}) \tag{5.9}$$

Even if this joint *pmf* is defined over discrete spaces, its estimation remains far from being tractable in practice (highly dimensional joint space). To make the computation of each perceptual task feasible we detail in this section a 2-steps procedure to compute joint *pmfs* of the form seen in equation (5.9). This 2 steps approach will be then used in the computation of the Localization, Association, Mapping, Velocity Estimation, Detection and Classification problems as detailed in section 5.4 of this chapter.

1st Step: Interactions with the other cells (Global Computation)

As mentioned above, because neighbouring cells are likely to be occupied by the same real object, the computation of $P(A_k(x_i)|Z_{0:k})$ has to be derived from the joint *pmf* below:

$$P(A_k(x_1), ..., A_k(x_N)|Z_{0:k})$$

The number of cells in the environment N is expected to be high, the estimation of this joint *pmf* is then a difficult problem. To solve this problem, we make a first assumption:

Joint probability mass function of the form $P(A_k(x_1), ..., A_k(x_N)|Z_{0:k})$ *can be approximated as follows:*

$$P(A_k(x_1), ..., A_k(x_N)|Z_{0:k}) \simeq \{ \prod_{1 \leq i \leq N} P_{\text{local}}(A_k(x_i)|Z_{0:k}) \} \Phi(A_k(x_1), ..., A_k(x_N), Z_{0:k}) \quad (5.10)$$

, where Φ *is a* $R^N \rightarrow R$ *function that penalizes unlikely configurations and* P_{local} *is the probability mass function related to a cell considered as independent from the others.*

The *pmf* related to each cell can then be obtained by marginalizing the above joint *pmf*.

$$P(A_k(x_i) = a_k(x_i)|Z_{0:k}) \simeq \sum_{a_k \in \mathcal{A}} \left\{ \left\{ \prod_{1 \leq j \leq N} P_{\text{local}}(A_k(x_j)|Z_{0:k}) \right\} \Phi(A_k(x_1), ..., A_k(x_N), Z_{0:k}) \right\} \quad (5.11)$$

, where $\mathcal{A}$ contains all the possible realisations of A_k such that $A_k(x_i) = a_k(x_i)$.

It is important to note that any available information can be used in the potential function Φ in addition to the random vector configuration a_k.

All the necessary interactions with other cells should then be incorporated in the potential function Φ allowing the *pmf* $P_{\text{local}}(A_k(x_j)|Z_{0:k})$ to be computed without taking care of the neighbouring cells. However, the computation of this local *pmf* should still take into account the other perceptual tasks.

2nd Step: Interactions with the other tasks (Local Computation)

We assume that the *pmf* $P_{\text{local}}(A_k(x_j)|Z_{0:k})$ should now be computed *locally* from the following simplified joint *pmf* to take into account the other perceptual tasks:

$$P_{\text{local}}(A_k(x_j) = a|Z_{0:k}) \simeq \sum P(A_k(x_j) = a, B_k(x_j), ...|Z_{0:k}) \quad (5.12)$$

, where B_k designates the random variable related to any other perceptual task.

The computation of the joint *pmf* (5.12) is still intractable in practice and further simplifications are needed. We propose to simplify the computation of this joint *pmf* by making the following second assumption.

Joint probability mass functions of the form $P(A_k(x_i), B_k(x_i)|Z_{0:k})$ can be correctly approximated by $P(A_k(x_i), B_{k-1}(x_i^a)|Z_{0:k})$ if the sensors acquisition period Δt is small compared to the time constant of the dynamical system related to B_k (x_i^a being the cell that was occupied at time $k-1$ by the object occupying the cell x_i at time k).

As a result, equation (5.12) can be rewritten as follows:

$$P_{\text{local}}(A_k(x_j) = a|Z_{0:k}) \simeq \sum P(A_k(x_j) = a, B_{k-1}(x_j^a), ...|Z_{0:k}) \tag{5.13}$$

This assumption is motivated by the fact that some of the *pmfs* described above are likely to evolve slowly over time. For example, the *pmf* related to object dynamic features at time $k-1$ can still be of great use to compute the detection *pmf* at time k. However, this requires to solve first a correspondence problem as an object point occupying a given cell (x_i^a) can potentially occupy another cell (x_i) at time k.

This former assumption implies that none of the perceptual tasks are computed simultaneously. The computation of each perceptual task *pmf* is addressed sequentially and makes use of all the other perceptual tasks *pmfs* already estimated (one step before if necessary). This approximation still allows a high level of interaction between the tasks.

Summary

Exact computations of the *pmfs* related to each perceptual task would lead directly to the computation of intractable joint *pmfs*. A 2 steps approach is proposed to compute a *pmf* of the form $P(A_k(x_i)|Z_{0:k})$.

First, this *global* computation is reduced to a *local* problem by modelling all the interactions between others cells through a potential function Φ that penalizes configurations that are globally unlikely.

Then, the local computation of $P_{\text{local}}(A_k(x_i)|Z_{0:k})$ is performed by integrating all the possible interactions with the other perceptual tasks. A perfect integration of these interactions would impose to address all the tasks simultaneously. Unfortunately, this implies, as seen above, to marginalize over another intractable joint *pmf*. That is why we propose instead to address each perceptual task sequentially. Each local computation being aided by all the other perceptual tasks *pmfs* already estimated.

5.4 The Algorithm in Practice

5.4.1 Introduction

We have only presented so far the basic principles of the proposed approach along with the necessary approximation to make them tractable in theory. We give now the outlines of the elegant perception algorithm that can be derived from these concepts in practice. This section is intended to meet two objectives.

1. Describe the chronological steps of this algorithm
2. Indicate for each step how the successive *local* an *global* computation of a given *pmf* can be implemented in practice.

The algorithm described in this chapter is not derived for any specific application nor any particular sensor. As a consequence, the practical functionality of every term derived in the following sections will be detailed, but no specific motion or measurement models will be described in this chapter as they depend mainly on applications. The implementation of this algorithm for the specific case of pedestrian perception is thoroughly detailed in chapter 6.

5.4.2 Step 0 - Initial Status

To start with, it is assumed that all the probability mass functions related to each perceptual tasks are available at time $k-1$. At time k new measurements z_k are received from the sensors. The knowledge stored at this point in the algorithm can be summed up as follows.

Measurements	z_k	Current
Localization	$P(L_{k-1} = l_j \| Z_{o:k-1})$	Previous
Association	$P(X_{k-2}^{next}(x_i) = x_j \| M_{k-2}(x_i) = 1, Z_{0:k-1})$	Previous
Mapping	$P(M_{k-1}(x_i) = 1 \| Z_{0:k-1})$	Previous
Dynamic Filtering	$P(V_{k-1}(x_i) = v_j \| M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous
Detection	$P(D_{k-1}(x_i, x_j) = 1 \| M_{k-1}(x_i) = 1, M_{k-1}(x_j) = 1, Z_{0:k-1})$	Previous
Classification	$P(C_{k-1}(x_i) = c_j \| M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous

5.4.3 Step 1 - Computation of the Localization problem

The computation of the Localization problem is different from the computation of the other tasks as it does not need to be done in every cell of the environment. Hence, the approximation mentioned above related to cells interactions does not need to be used. It is besides important to address the Localization problem as the first step of this algorithm. Indeed, new sensor observations will only be correctly used if the new location of the sensing platform is correctly estimated. The situation that has to be disambiguated by the Localization computation can be schematized as shown in figure 5.6.

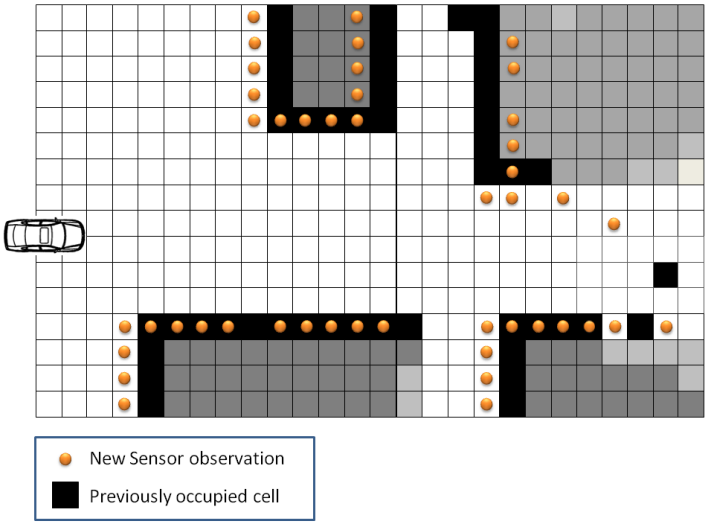


Figure 5.6: Schematic view of the information available at this point of the process. The localization of the sensing platform needs to be inferred.

The computation of the Localization *pmf* can be decomposed as an usual Bayesian filtering problem:

$$P(L_k|Z_{0:k}) = \eta \underbrace{P(z_k|L_k, Z_{0:k-1})}_{\text{Correction Term}} \underbrace{P(L_k|Z_{0:k-1})}_{\text{Prediction Term}} \tag{5.14}$$

The prediction term can be directly computed from the previous localization *pmf* and the motion model of the sensing platform:

$$P(L_k|Z_{0:k-1}) = \sum_{l_k \in E \times V} \underbrace{P(L_k|L_{k-1} = l_j, Z_{0:k-1})}_{\text{Motion Model}} \underbrace{P(L_{k-1} = l_j|Z_{0:k-1})}_{\text{Previous Localization } pmf} \tag{5.15}$$

It is interesting to note that the motion model can be further derived to incorporate information about the previous Mapping and Detection *pmfs* for example. This would allow to use a motion model that takes into account the possible interactions between the sensing platform and the objects in the environment.

The correction term should be derived to incorporate the information available about the currently occupied cells. Unfortunately, this knowledge is only available for time $k-1$. As computing a prediction of a whole map is usually not easy, only the cells that have been classified as static may be incorporated in this computation. This can be done by using the available information (*pmfs*) related to Classification and Velocity Estimation. The resulting term would then take the following form:

$$P(z_k|L_k, Z_{0:k-1}) = \sum_{c_{k-1}, v_{k-1}, m_{k-1}} \underbrace{P(z_k|C_{k-1}, V_{k-1}, M_{k-1}, L_k)}_{\text{Observation Likelihood}} \underbrace{P(C_{k-1}|, V_{k-1}, M_{k-1}, Z_{0:k-1})}_{\text{Previous Classification } pmf} \underbrace{P(V_{k-1}|M_{k-1}, Z_{0:k-1})}_{\text{Previous Velocity } pmf} \underbrace{P(M_{k-1}|Z_{0:k-1})}_{\text{Previous Mapping } pmf} \tag{5.16}$$

To alleviate the computational burden that such a summation represents, some of these *pmfs* can be approximated by their most probable values (*maximum a posteriori* estimate).

Localizing the platform from the surrounding moving objects is possible in theory if their velocity in the street reference frame is known precisely. But in practice, precise localization (without using *GPS*) can only be computed if static obstacles are seen regularly.

Summary

At the end of this first step, a hopefully correct new Localization *pmf* is available. This new information can be depicted as shown in figure 5.7. Note that the Localization information allows also to roughly align the new sensors measurements with the former occupancy map.

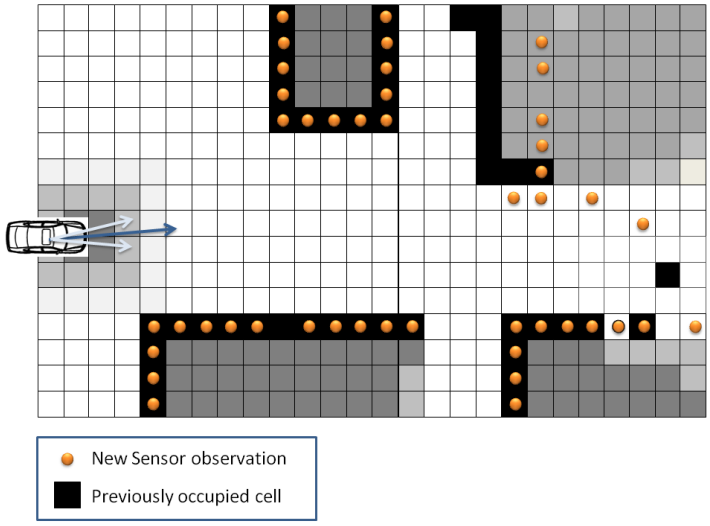


Figure 5.7: Schematic view of the Localization information available at this point of the process.

Interactions between the perceptual tasks that are potentially used in this computation can be schematized as described in figure 5.8. Available *pmfs* at this point can be summed up as follows.

Measurements	z_k	Current
Localization	$P(L_k = l_j \| Z_{o:k})$	Current
Association	$P(X_{k-2}^{next}(x_i) = x_j \| M_{k-2}(x_i) = 1, Z_{0:k-1})$	Previous
Mapping	$P(M_{k-1}(x_i) = 1 \| Z_{0:k-1})$	Previous
Velocity Estimation	$P(V_{k-1}(x_i) = v_j \| M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous
Detection	$P(D_{k-1}(x_i, x_j) = 1 \| M_{k-1}(x_i) = 1, M_{k-1}(x_j) = 1, Z_{0:k-1})$	Previous
Classification	$P(C_{k-1}(x_i) = c_j \| M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous

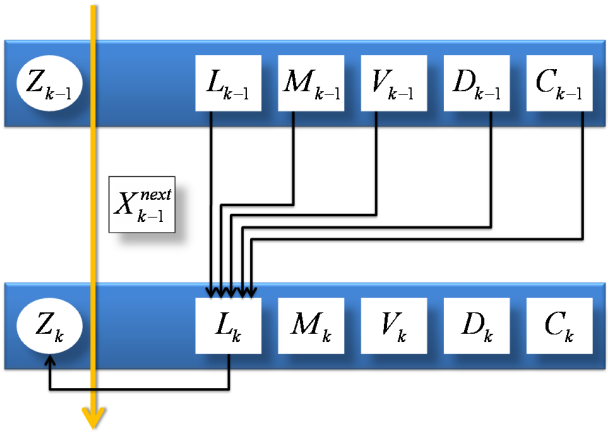


Figure 5.8: Dependencies with other perception tasks used in the localization computation.

5.4.4 Step 2 - Computation of the Association problem

Objects in the environment are all potentially moving. As a result, an object point occupying cell A at time $k-1$ is likely to be located in an other cell B at time k. Solving this correspondence problem is crucial as lot of information can be extracted from it.

To solve this problem we use a greedy approach that is well adapted to grid-based representation. For any given cell whose occupancy probability is not null, every possible association is first investigated locally before being integrated globally using a potential function that penalizes unlikely global association.

Local computation

A first *local* computation of where a point occupying a cell at time $k-1$ might be at time k is performed using the available *pmfs*. This corresponds to the computation of the following intermediate *pmf*:

$$P_{\text{local}}(X_{k-1}^{next}(x_j) | M_{k-1}(x_j) = 1, Z_{0:k})$$

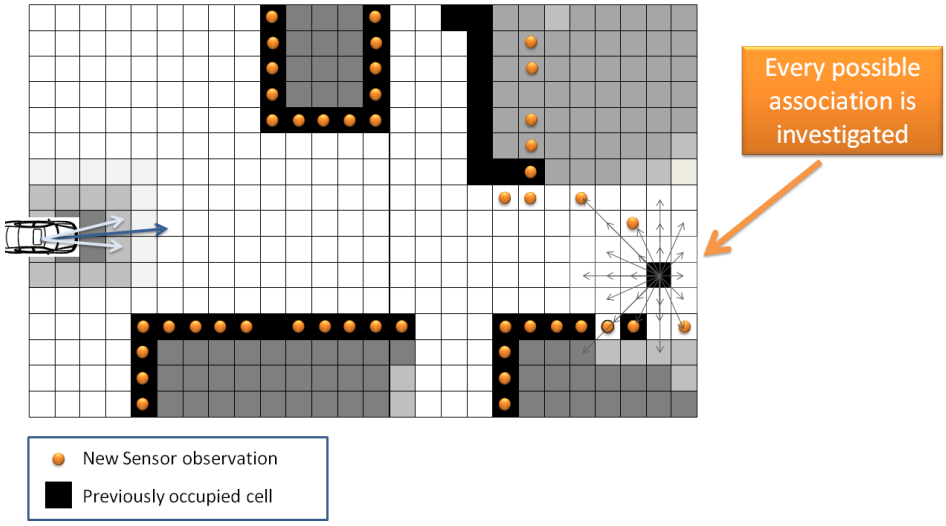


Figure 5.9: Schematic view of the greedy approach used to solve the Association problem.

As mentioned above, the subscript *local* in this probability indicates that the computation is temporarily made without taking care of surrounding cells (*local* computation). This *local* computation is based on an usual Bayesian filtering equation where a prediction is computed first and then corrected by the current sensor observations.

$$P_{\text{local}}(X_{k-1}^{next}(x_j)|M_{k-1}(x_j)=1,Z_{0:k}) = \eta \underbrace{P_{\text{local}}(z_k|X_{k-1}^{next}(x_j),M_{k-1}(x_j)=1,Z_{0:k-1})}_{\text{Local Correction}} \underbrace{P_{\text{local}}(X_{k-1}^{next}(x_j)|M_{k-1}(x_j)=1,Z_{0:k-1})}_{\text{Local Prediction}} \quad (5.17)$$

A good prediction of where a given cell x_i at time $k-1$ might go at time k can be computed using the available information about the speed and the class of the point occupying that cell. This can be done by using an appropriate motion model that takes these two parameters into account. Because the velocity and the classification information are available as *pmfs*, the computation of this prediction term should be performed through an appropriate marginalization over all the possible velocities and classes for the given cell.

The local correction term is derived using an appropriate sensor model and the Localization *pmf* computed at step 1. Indeed, to locally know which sensor measurement should be used for the update, the information about the current Localization is needed. Besides, the sensor model is expected to put the sensor information into a probabilistic form. Such sensor models are discussed in (Leal, 2003) for example.

Interaction with other cells

Such a *local* computation is of course insufficient to solve correctly the global Association problem but is still useful to compute the final Association *pmf*. As mentioned in section 5.3, the final Association *pmf* can be approximated by the following equation.

$$P(X_{k-1}^{next}(x_i) = \hat{x} | M_{k-1}(x_i) = 1, Z_{0:k}) \simeq$$
$$\sum_{a_k \in \mathcal{A}} \left\{ \left\{ \prod_{1 \leq j \leq N} P_{\text{local}}(X_{k-1}^{next}(x_j) | M_{k-1}(x_i) = 1, Z_{0:k}) \right\} \Phi_{\text{association}}(a_k, E, Z_{0:k}) \right\} \quad (5.18)$$

, where $\mathcal{A}$ is the set of all the possible global associations such that $X_{k-1}^{next}(x_i) = \hat{x}$.

This procedure is an efficient way to correct *local* computations through the use of a *global* function $\Phi_{\text{registration}}$ that penalizes globally impossible or conflicting associations. This potential function should of course depends directly on the proposed global configuration a_k but can also uses any other information available at this point of the process.

This function can be designed to penalize the global configurations that are not consistent with the following heuristics:

- Two points belonging to the same incompressible object should stay at the same distance from one another.
- Two points belonging to different objects should not converge to the same cell.
- The association of the points with new sensor measurements should be able to "explain" the maximum number of new sensor observations.

It is important to note that the two first heuristics are based on Detection and Classification information that is available at this point of the process as two *pmfs* computed at time $k-1$.

Summary

At the end of this second step, a new Association *pmf* is available that can be schematized as seen in picture 5.10 (using the corresponding *MAP* estimates).
The available *pmfs* at this point of the algorithms can be summarized as follows.

Measurements	z_k	Current
Localization	$P(L_k = l_j \| Z_{o:k})$	Current
Association	$P(X_{k-1}^{next}(x_i) = x_j \| M_{k-1}(x_i) = 1, Z_{0:k})$	Current
Mapping	$P(M_{k-1}(x_i) = 1 \| Z_{0:k-1})$	Previous
Velocity Estimation	$P(V_{k-1}(x_i) = v_j \| M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous
Detection	$P(D_{k-1}(x_i, x_j) = 1 \| M_{k-1}(x_i) = 1, M_{k-1}(x_j) = 1, Z_{0:k-1})$	Previous
Classification	$P(C_{k-1}(x_i) = c_j \| M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous

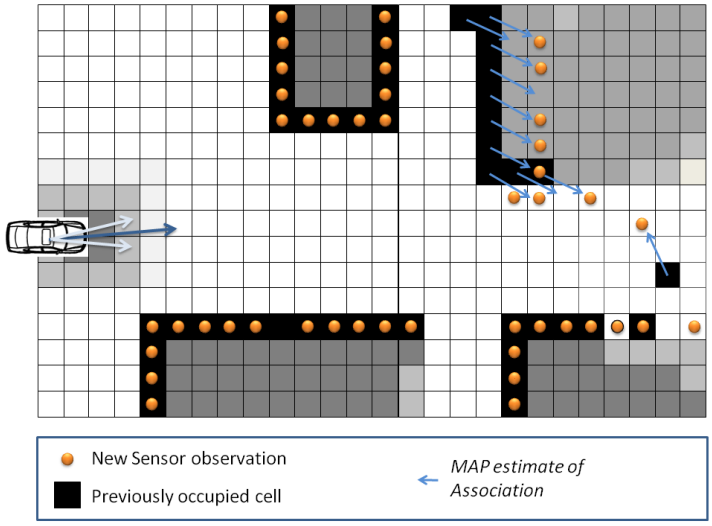


Figure 5.10: Schematic view of the new Association *pmf* represented here as most probable estimates for clarity.

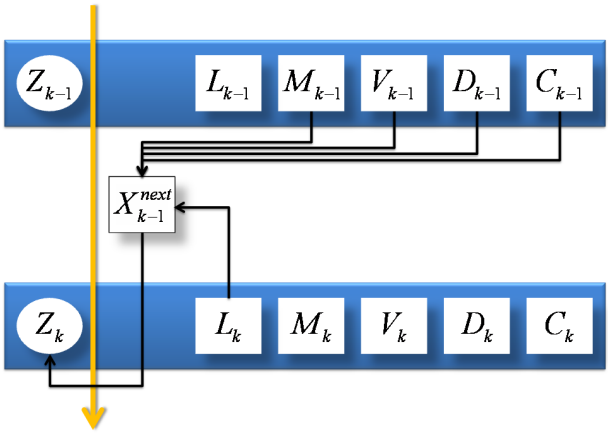


Figure 5.11: Dependencies with other perception tasks used in the Association problem computation.

The computation of the Association problem is thus potentially based on all these available *pmfs*. These interactions can be schematized as shown in figure 5.11

5.4.5 Step 3 - Computation of the Mapping problem

Since the Association problem is solved, it now possible to compute the new occupancy probability of every cell. As before, this computation is first performed locally without taking into account the interactions with the neighbouring cells.

Local computation

If the cell x_i is occupied at time k, this event has two possible causes: the point located in cell x_i has already been seen before or it is a new point that is seen for the first time. In the following we note $S_k(x_i)$ the random variable such that:

$$S_k(x_i) = \begin{cases} 1 & \text{if "the point located in } x_i \text{ at time } k \text{ (if it exists) has already been seen" is true} \\ 0 & \text{else} \end{cases}$$

By using this notation, the *local* probability for a cell to be occupied at time k can be decomposed as follows.

$$\begin{aligned} &P_{\text{local}}(M_k(x_i) = 1 | Z_{0:k}) = \\ &\quad \underbrace{P_{\text{local}}(M_k(x_i) = 1 | S_k(x_i) = 1, Z_{0:k})}_{\text{Occupancy if already seen}} \ \underbrace{P(S_k(x_i) = 1 | Z_{0:k})}_{\text{Status of the cell}} \\ &\quad + \underbrace{P_{\text{local}}(M_k(x_i) = 1 | S_k(x_i) = 0, Z_{0:k})}_{\text{Occupancy if never seen}} \ P(S_k(x_i) = 0 | Z_{0:k}) \end{aligned} \tag{5.19}$$

Status of the cell

This probability can be computed from the available Association *pmf*. Indeed, if a cell x_i contains a point that has already been seen before then this point must have been registered during the association process performed in step 2. This information can be extracted from the Association *pmf* in many ways but the following solution proved to be efficient in practice:

$$\begin{aligned} &P(S_k(x_i) = 1 | Z_{0:k}) = \\ &\quad \begin{cases} \gamma & \text{if } \exists x_j \in E \text{ st } x_i = \arg\max_{x_l} P(X_{k-1}^{next}(x_j) = x_l | M_{k-1}(x_j) = 1, Z_{0:k}) \\ 1 - \gamma & \text{else} \end{cases} \end{aligned} \tag{5.20}$$

, with $\gamma \in [0, 1]$.

This equation makes use of a simple heuristic: only the current cells that correspond to the most probable association with a previous cell are given a high probability to have already been seen before.

Occupancy if already seen

If the point that is located in a given cell has already been seen before, then the Association *pmf* contains information about where that point was located before. As a consequence, by summing over all the possible cells that this point might have occupied before (weighted by the appropriate association probability), it is possible to infer the occupancy probability of this cell from the occupancy probability of the previous cells. This process is depicted on figure 5.12.

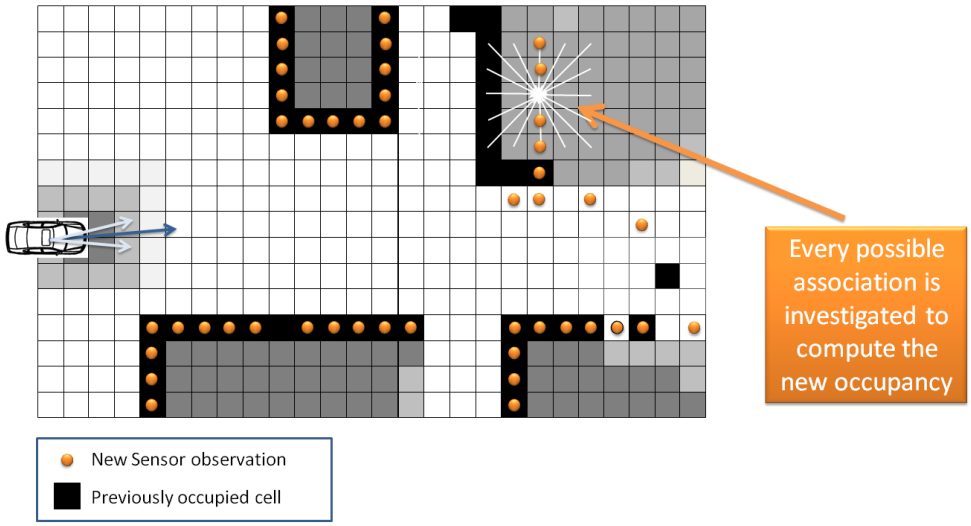


Figure 5.12: If a point occupying a cell has already been seen before, all the possible previous cells where this point might have been are investigated to compute the new occupancy probability of that cell.

Occupancy if never seen

When a point occupying a cell is seen for the first time, the occupancy of that cell is naturally directly computed from the sensor observation. This requires first to transform the sensors observations as an occupancy observation. This is usually called a *sensor occupancy model*. Further details about occupancy models can be found in (Leal, 2003). This computation also requires information about the sensing platform localization to use the appropriate sensor observation for a given cell. This information can of course be extracted from the Localization *pmf* estimated in step 1 of this algorithm.

Interaction with other cells

Taking into account the interactions with the other cells is not critical in mapping computation. In fact, most of the important interactions between cells are already embodied in the Association *pmf* that is used to solve mapping. However, it is still possible to make use of the information contain in the Detection and Classification *pmf* (estimated at time $k-1$ here) to refine the map of some object. It would for example be possible to increase the occupancy probability of cells that are located inside what has been classified as a building or a vehicle.

Summary

At the end of this process, the new occupancy map of the environment is available. It is critical to note that this approach allow to both map static and moving objects as schematized in figure 5.13. The interactions with the other tasks that are potentially used in the mapping computation are depicted in figure 5.14. Finally, available *pmfs* at this point can be summarized as shown below.

Measurements	z_k	Current
Localization	$P(L_k = l_j \vert Z_{o:k})$	Current
Association	$P(X_{k-1}^{next}(x_i) = x_j \vert M_{k-1}(x_i) = 1, Z_{0:k})$	Current
Mapping	$P(M_k(x_i) = 1 \vert Z_{0:k})$	Current
Velocity Estimation	$P(V_{k-1}(x_i) = v_j \vert M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous
Detection	$P(D_{k-1}(x_i, x_j) = 1 \vert M_{k-1}(x_i) = 1, M_{k-1}(x_j) = 1, Z_{0:k-1})$	Previous
Classification	$P(C_{k-1}(x_i) = c_j \vert M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous

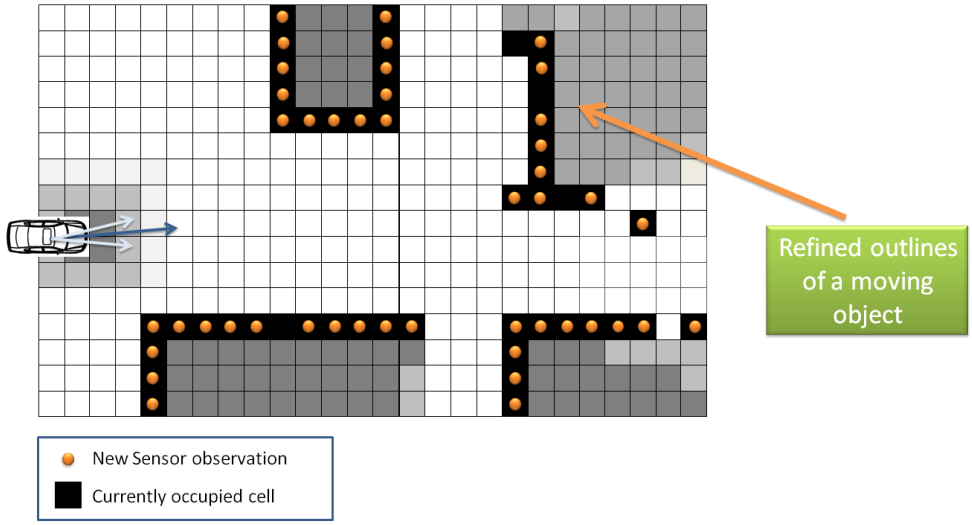


Figure 5.13: The proposed approach is able to map both static and moving objects as depicted here.

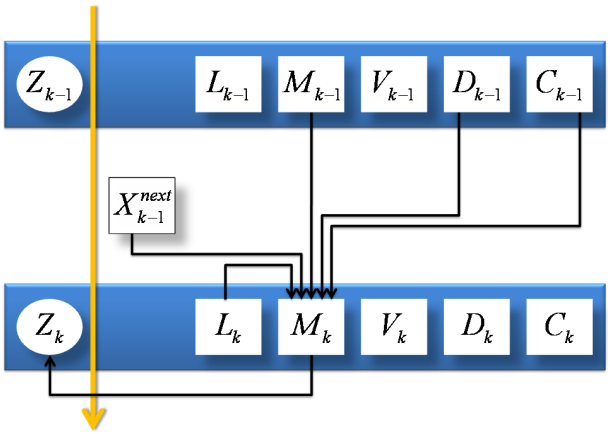


Figure 5.14: Dependencies with other perception tasks used in the mapping computation.

5.4.6 Step 4 - Computation of the Velocity Estimation problem

The dynamical features of the points occupying every cells can be now be estimated. Note that although this problem is called for simplicity the *Velocity Estimation problem*, all relevant features related to the dynamic behavior of a point occupying a cell can be estimated here (such accelerations, etc...).

Local computation

By considering the computation of the Velocity Estimation problem in a cell as independent from the similar computation in the neighbouring cells, a good estimate of the velocity of a given cell can be obtained from the association result. Indeed, if the previous location of a point occupying a cell is known, information about its speed can be inferred. However, a given cell might also be occupied by an object point that is seen for the first time, in that case no information can be extracted from the Association *pmf*. This *local* computation can then be decomposed in the same way as the computation of the Mapping problem described above.

$$\begin{aligned} P_{\text{local}}(V_k(x_i)|M_k(x_i)=1,Z_{0:k}) = & \\ \underbrace{P_{\text{local}}(V_k(x_i)|S_k(x_i)=1,M_k(x_i)=1,Z_{0:k})}_{\text{Velocity if already seen}} & \underbrace{P(S_k(x_i)=1|Z_{0:k})}_{\text{Status of the cell}} \\ + \underbrace{P_{\text{local}}(V_k(x_i)|S_k(x_i)=0,M_k(x_i)=1,Z_{0:k})}_{\text{Velocity if never seen}} & P(S_k(x_i)=0|Z_{0:k}) \end{aligned} \tag{5.21}$$

Status of the cell

Estimating this probability is a problem that has already been discussed in the computation of the new occupancy map. The same solutions can naturally be used.

Velocity if already seen

As stated above, when the point located in x_i is considered as already seen, its velocity (and acceleration) can be directly derived from the velocity of the cell where this point was before. Of course, this estimation should be performed by marginalizing over all the possible association and velocity hypothesis. This strategy is depicted in figure 5.15. It should be noted that any sensor observation related to objects dynamic features can also be integrated in this computation. Similarly, the Classification *pmf* estimated at time $k-1$ can be used to feed the motion model that this computation requires.

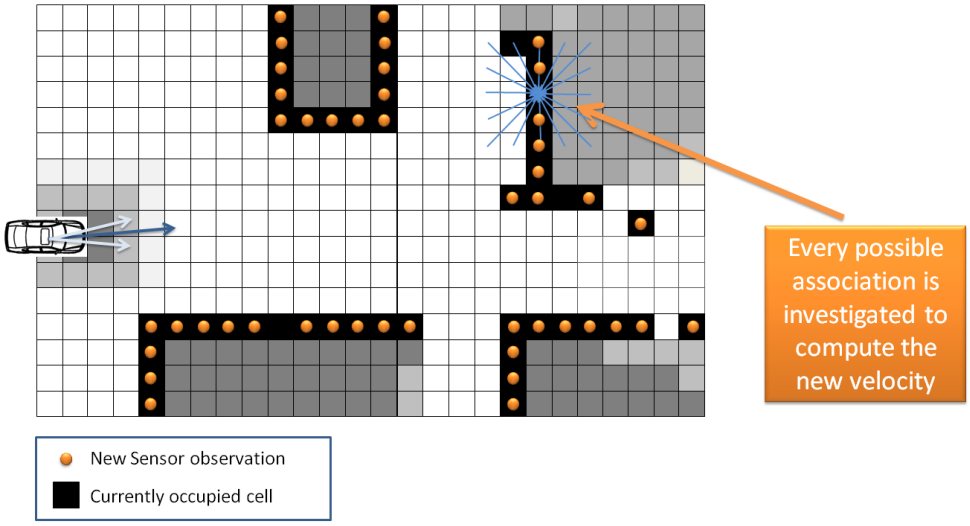


Figure 5.15: If a point occupying a cell has already been seen, its velocity can be derived by investigating all the possible association hypothesis with previous cells.

Velocity if never seen

If the point located in a given cell is seen for the first time, the above strategy cannot be used. If sensors are able to collect data about objects speed or acceleration, this information can be used to compute this term (through the Localization *pmf*). Else, a non informative prior can be used. In chapter 6, the following prior is used:

$$P_{\text{local}}(V_k(x_i)|S_k(x_i)=0, M_k(x_i)=1, Z_{0:k}) = \frac{1}{\text{card}(V)} \tag{5.22}$$

Interaction with other cells

Taking into account the interaction is not critical here to compute relevant velocities. Indeed, as this computation is mainly based on the Association *pmf*, most of the interactions have already been taken into accounts. However, the detection and classification information (at

time $k-1$) could be used to homogenize velocities over the different points of a same given object.

Summary

The process described in this paragraph is intended to produce a relevant map of the velocities in the environment as shown in figure 5.16. Interactions that can potentially be used in this computation are depicted in figure 5.17. Finally, the set of probability mass functions that are available at this point of the process can be summarised as seen in the table below.

Measurements	z_k	Current
Localization	$P(L_k = l_j \mid Z_{o:k})$	Current
Association	$P(X_{k-1}^{next}(x_i) = x_j \mid M_{k-1}(x_i) = 1, Z_{0:k})$	Current
Mapping	$P(M_k(x_i) = 1 \mid Z_{0:k})$	Current
Velocity Estimation	$P(V_k(x_i) = v_j \mid M_k(x_i) = 1, Z_{0:k})$	Current
Detection	$P(D_{k-1}(x_i, x_j) = 1 \mid M_{k-1}(x_i) = 1, M_{k-1}(x_j) = 1, Z_{0:k-1})$	Previous
Classification	$P(C_{k-1}(x_i) = c_j \mid M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous

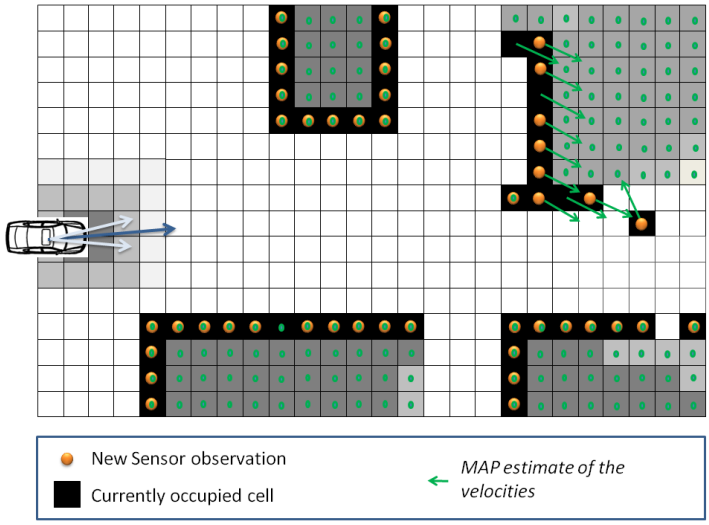


Figure 5.16: Schematic view of the velocity *pmf* that is computed in this section. Only the most probable estimates are represented here for clarity reasons.

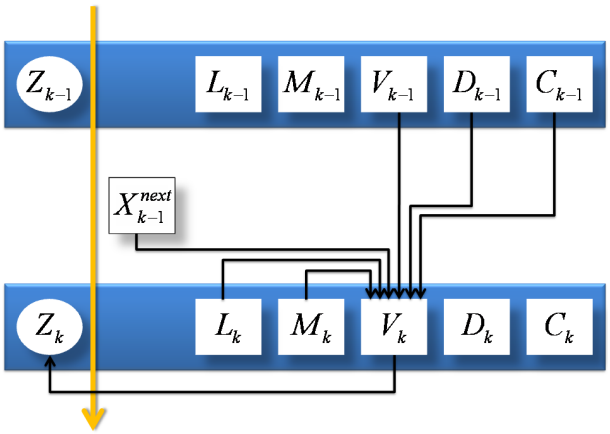


Figure 5.17: Dependencies with other perception tasks used in the computation of the Velocity Estimation problem.

5.4.7 Step 5 - Computation of the Detection problem

At this point of the process, the information that is necessary to compute accurately the Detection problem is available. Indeed, velocity (acceleration) estimates are available for any given cell (provided that its probability of being occupied is not equal to zero). An accurate Detection based both on geometric and dynamic criteria is then possible.

Local computation

Let us introduce for simplicity the random variable $S_k(x_i, x_j)$ and $M(x_i, x_j)$ such that:

$$S_k(x_i, x_j) = \begin{cases} 1 & \text{if } S_k(x_i) = S_k(x_j) = 1 \\ 0 & \text{else} \end{cases} \tag{5.23}$$

$$M_k(x_i, x_j) = \begin{cases} 1 & \text{if } M_k(x_i) = M_k(x_j) = 1 \\ 0 & \text{else} \end{cases} \tag{5.24}$$

As for the computations of most of the other tasks, a distinction has to be made based on whether the two cells that are considered (x_i and x_j) have already been seen before or not.

$$P_{\text{local}}(D_k(x_i,x_j)=1|M_k(x_i,x_j)=1,Z_{0:k}) =$$

$$\underbrace{P_{\text{local}}(D_k(x_i,x_j)=1|S_k(x_i,x_j)=1,M_k(x_i,x_j)=1,Z_{0:k})}_{\text{Detection if already seen}} \quad \underbrace{P(S_k(x_i,x_j)=1|Z_{0:k})}_{\text{Status of the cells}}$$
$$+$$
$$\underbrace{P_{\text{local}}(D_k(x_i,x_j)=1|S_k(x_i,x_j)=0,M_k(x_i,x_j)=1,Z_{0:k})}_{\text{Detection if never seen}} \quad P(S_k(x_i,x_j)=0|Z_{0:k})$$

Status of the cells

The estimation of this probability can be performed in many ways. In chapter 6, it is assumed that this estimation can be performed as if $S_k(x_i)$ and $S_k(x_j)$ were independent $\forall(x_i,x_j)\in E^2$ although there are obviously not. This leads to the following equation:

$$P(S_k(x_i,x_j)=1|M_k(x_i,x_j)=1) = P(S_k(x_i)=1|M_k(x_i)=1)\times P(S_k(x_j)=1|M_k(x_j)=1) \quad (5.25)$$

, where the terms of the product are computed as discussed in the previous steps of this algorithm. This assumption is not correct but turned out to be a satisfactory approximation in practice.

Detection if already seen

When the two points related to the two given cells (x_i and x_j) have already been seen, the local detection probability can be computed from a combination of four different sources.

1. The information contained in the current occupancy "map" (geometric based detection). Indeed two occupied cells that are close from one another are likely to belong to the same object.

2. The information contained in the velocity "map" computed at step 4 (dynamic based detection). Two points occupying two cells and having similar velocities are likely to belong to the same object even if they are not very close.

3. The information contained in the Detection *pmf* estimated at time $k-1$. Two points that belong to the same object at time $k-1$ are likely to belong to the same object at time k.

4. The information contained in the Classification *pmf* estimated at time $k-1$. Two points that were classified as being parts of a vehicle are likely to belong to the same vehicle.

This information is available as four different *pmfs* and should consequently be incorporated as weighted sums (marginalization) into the computation of this term.

Detection if never seen

If one of the points occupying the two given cells (x_i and x_j) is seen for the first time, most of the above information sources are useless. In this case, the Detection is only based on geometric criteria extracted from the current occupancy map (Mapping *pmf*).

Interaction with the other cells

As the computation of the local Detection *pmf* already involves two cells, some of this interaction is already managed. However, it might be still useful to make use of a potential function $\Phi_{\text{detection}}$ to homogenize globally the detection *pmf* computed locally. Indeed, if cell A is likely to belong to the same object as cell B, and if cell A is also likely to belong to the same object as cell C, then cell A and cell C are likely to belong to the same object. This cannot be ensured by only computing the Detection *pmf* locally.

Summary

The current Detection probability mass function is now available. This computation implies potentially several other perceptual tasks as shown in figure 5.19. The information available at this point can be summarized in the table below.

Measurements	z_k	Current
Localization	$P(L_k = l_j \vert Z_{o:k})$	Current
Association	$P(X_{k-1}^{next}(x_i) = x_j \vert M_{k-1}(x_i) = 1, Z_{0:k})$	Current
Mapping	$P(M_k(x_i) = 1 \vert Z_{0:k})$	Current
Velocity Estimation	$P(V_k(x_i) = v_j \vert M_k(x_i) = 1, Z_{0:k})$	Current
Detection	$P(D_k(x_i, x_j) = 1 \vert M_k(x_i) = 1, M_k(x_j) = 1, Z_{0:k})$	Current
Classification	$P(C_{k-1}(x_i) = c_j \vert M_{k-1}(x_i) = 1, Z_{0:k-1})$	Previous

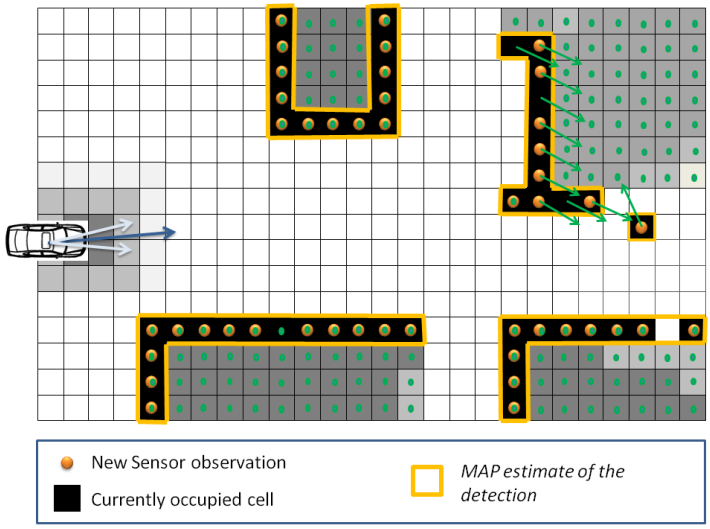


Figure 5.18: Schematic view of the Detection *pmf* computed in this section. Only the most probable estimates are represented here for clarity reasons.

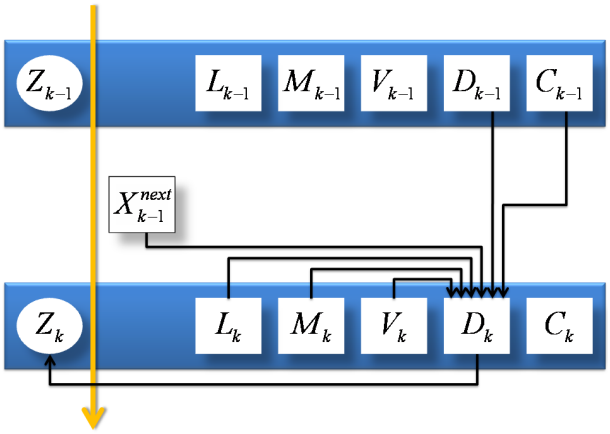


Figure 5.19: Dependencies with other perception tasks used in the computation of the Detection problem.

5.4.8 Step 6 - Computation of the Classification problem

The last step of the proposed algorithm is the computation of the Classification problem. Interactions between cells are crucial in the computation of this problem. Indeed, a cell can not be classified as being part of a "vehicle" if the surrounding cells are not classified accordingly. But it is still possible and potentially useful to perform a first computation locally.

Local computation

Without taking into account the neighbouring cells, the classification of a given cell x_i can still be inferred from the various *pmfs* that are available at this point in the algorithm. However, this should be decomposed first depending on whether the point occupying that cell is seen for the first time or has already been seen.

$$\begin{aligned} P_{\text{local}}(C_k(x_i)|M_k(x_i)=1,Z_{0:k}) = & \\ \underbrace{P_{\text{local}}(C_k(x_i)|S_k(x_i)=1,M_k(x_i)=1,Z_{0:k})}_{\text{Classification if already seen}} & \underbrace{P(S_k(x_i)=1|Z_{0:k})}_{\text{Status of the cell}} \\ + \underbrace{P_{\text{local}}(C_k(x_i)|S_k(x_i)=0,M_k(x_i)=1,Z_{0:k})}_{\text{Classification if never seen}} & P(S_k(x_i)=0|Z_{0:k}) \end{aligned} \tag{5.26}$$

Status of the cells

The computation of this probability is already discussed in the other sections of this chapter.

Classification if already seen

When the point occupying a given cell x_i has already been seen before, then a combination of the three following information sources is possible.

1. The information contained in the current Velocity Estimation *pmf* for the considered cell gives information about the possible objects classes the point occupying that cell might belong to.

2. The information contained in the Classification *pmf* estimated at time $k-1$ and accessible through the Association *pmf* can also be useful. Indeed, a point classified as belonging to a vehicle at time $k-1$ is likely to belong to the same class of objects at time k.

3. The sensor can directly make observations about the class of an object. This information accessible through the Localization *pmf* can be integrated to this local computation. Cameras can indeed provide information about objects colors into a given area (a given cell here) that could be incorporated in this way.

Classification if never seen

If the object point occupying a cell has never been seen before, most of the above information are still usable. Indeed, provided that the velocity estimate corresponding to that point is relevant (directly measured by the sensor for example), this information can be used to infer locally a classification. Sensors observations can also be integrated as mentioned above. However, because this point seen for the first time has not been registered yet, no information from time $k-1$ can be used.

Interaction with other cells

The local computation of the classification problem is useful but not sufficient. Indeed, a cell cannot be appropriately classified without looking at the neighbouring cells. Most existing classification algorithms are indeed based on objects and their use should be made possible. To do so, an object-based representation has to be derived from our framework. Passing from a grid-based representation to an object-based representation should be performed while retaining a high level of uncertainty modelling.

In our framework an object at time k can be described as a set of cells that satisfy the following condition.

$$\mathbb{O}_k^{\delta}(x_i) = \{x_j \in E \setminus P(D_k(x_i, x_j) = 1 | M_k(x_i, x_j) = 1, Z_{0:k}) \geq \delta\} \tag{5.27}$$

Depending on the value chosen for δ, an object $\mathbb{O}_k^{\delta}(x_i)$ defined by this equation will be more or less likely to correspond to a real object in the scene. It is possible to express this probability as the following product:

$$P(\mathbb{O}_k^{\delta}(x_i) \text{ is real} | Z_{0:k}) = \prod_{(x_i,x_j)\in E^2} P(D_k(x_i, x_j) = 1 | M_k(x_i, x_j) = 1, Z_{0:k}) \\ P(M_k(x_i) = 1 | Z_{0:k}) \;\; P(M_k(x_j) = 1 | Z_{0:k}) \tag{5.28}$$

Equations (5.27) and (5.28) offer an elegant way to pass naturally from a grid-based representation to an object based representation while retaining a high level of uncertainty representation.

To compute the *global* Classification *pmf* for any given cell $x_i \in E$, the method proposed in section 5.3 is used as follows.

$$P(C_k(x_i) = \hat{c} | M_k(x_i) = 1, Z_{0:k}) \simeq \\ \sum_{a_k \in \mathcal{A}} \left\{ \left\{ \prod_{1\leq j\leq N} P_{\text{local}}(C_k(x_j) | M_k(x_i) = 1, Z_{0:k}) \right\} \Phi_{\text{Classification}}(a_k) \right\} \tag{5.29}$$

, where $\mathcal{A}$ is the set of all the possible global Classification configurations such that $C_k(x_i) = \hat{c}$.

The potential function $\Phi_{\text{Classification}}$ is intended to penalize the classification configurations that are globally unlikely. This can be performed by any existing classification algorithm to which appropriate objects are given. This could be implemented through the following potential function:

$$\Phi_{\text{Classification}}(a_k) = \prod_{x_i \in E} \underbrace{P(C_k(x_i) = c_i | \mathbb{O}_k^{\delta}(x_i) \text{ is real}, Z_{0:k})}_{\text{Computed by an object-based classification algorithm}} \tag{5.30}$$

, with a value of δ being chosen to feed the object-based classification algorithm with a realistic object. However, this process does not maintain any uncertainties about the object that is sent to the object-based classification algorithm. Any badly identified object can prevent the process to perform relevant Classification.

This potential function should be adapted to make use of the uncertainty information available for each object. This can be done by the following modified function where a marginalization is performed over all the possible objects hypothesis.

$$\Phi_{\text{Classification}}(a_k) =$$

$$\prod_{x_i \in E} \beta(x_i) \sum_{\delta_i \in \Delta} \underbrace{P(C_k(x_i) = c_i | \mathbb{O}_k^{\delta_i}(x_i) \text{ is real}, Z_{0:k})}_{\text{Computed by an object-based classification algorithm}} \quad P(\mathbb{O}_k^{\delta_i}(x_i) \text{ is real} | Z_{0:k}) \tag{5.31}$$

, with $\beta(x_i) = (\sum_{\delta_i \in \Delta} P(\mathbb{O}_k^{\delta_i}(x_i) \text{ is real} | Z_{0:k}))^{-1}$ and $\Delta = \{\delta_1, ..., \delta_m\} \in [0,1]^m$.

This former potential function formulation requires more computational power but is a way to maintain a high level of uncertainty management. It should be noted that any *pmf* currently available can be given to the object-based classification algorithm if necessary.

Summary

At the end of this sixth step, a new Classification *pmf* is now available. This *pmf* can be depicted as seen in figure 5.20 and the interactions with other tasks that this computation might use are summarized in figure 5.21.

Measurements	z_k	Current
Localization	$P(L_k = l_j \| Z_{o:k})$	Current
Association	$P(X_{k-1}^{next}(x_i) = x_j \| M_{k-1}(x_i) = 1, Z_{0:k})$	Current
Mapping	$P(M_k(x_i) = 1 \| Z_{0:k})$	Current
Velocity Estimation	$P(V_k(x_i) = v_j \| M_k(x_i) = 1, Z_{0:k})$	Current
Detection	$P(D_k(x_i, x_j) = 1 \| M_k(x_i) = 1, M_k(x_j) = 1, Z_{0:k})$	Current
Classification	$P(C_k(x_i) = c_j \| M_k(x_i) = 1, Z_{0:k})$	Current

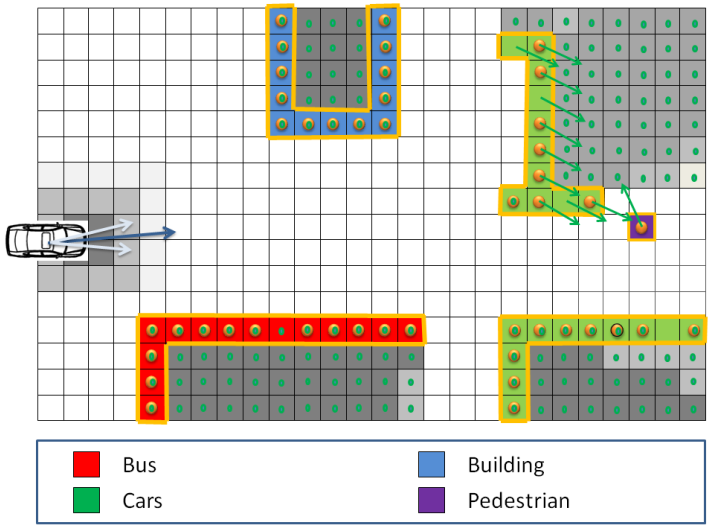


Figure 5.20: Schematic view of the Classification *pmf* computed in this section. Only the most probable estimates are represented here.

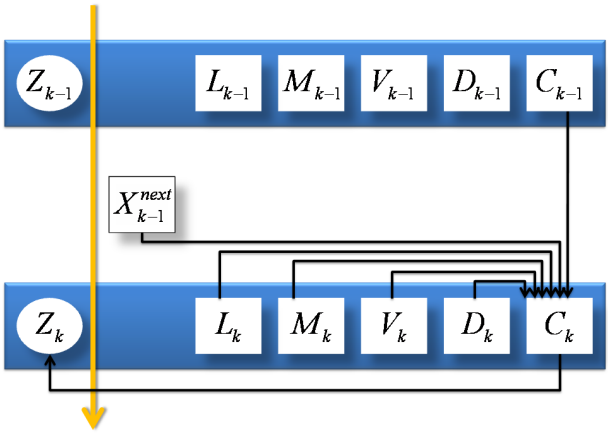


Figure 5.21: Dependencies with other perception tasks used in the computation of the Classification problem.

5.5 Conclusion

In this chapter, an original algorithm is proposed to solve sequentially all the perceptual tasks. This perception algorithm is based on an unified and powerful grid-based representation that allows the efficient representation of all the uncertainty generated by the system as probability mass functions (*pmfs*). Besides, in this framework, any given perceptual task can be solved sequentially as an usual discrete inference problem where any relevant information coming from the other tasks can be elegantly introduced.

The main strength of this approach is certainly its capacity to embed all the facets of the perception problem (Detection, Tracking, Classification, Mapping and Localization) into the same mathematical formalism. This unified mathematical approach is then an elegant way to enable any required interaction between two perceptual tasks. As a result, the algorithm described in this chapter is able to simultaneously solve all the critical issues of common approaches discussed in Chapter 4. This particular point will be proven on real data in Chapter 6.

The main weakness of this perception algorithm however is certainly the inherent computational burden involved by most of its steps. But this computational load can be greatly soften while still preserving most of the interesting properties of this approach as shown in the next chapter.

Résumé en français du chapitre 6

Dans le but d'évaluer la validité des concepts théoriques présentés au chapitre 5, nous détaillons dans ce chapitre un deuxième système de perception de piétons directement basé sur l'approche présentée au chapitre précédent. Une partie importante de ce chapitre est consacrée à la comparaison des performances de ce système avec celui présenté au chapitre 3 sur le critère qui nous intéresse ici : la fiabilité.

Chapter 6

Pedestrian Perception - Reliable System

Contents

6.1 Introduction

In this chapter, we present an attempt to adapt the grid-based approach presented in Chapter 5 to the specific problem of pedestrian perception with a ladar. The framework presented in the previous chapter is powerful but can imply heavy computational requirements depending on the level of interaction that is implemented.

For any given application, it is in fact possible to adapt the algorithm for reasonable execution time while still retaining most of its benefits. **This can be made for example by only implementing interactions that are especially needed**. In Chapter 4, three major interactions have been identified as important to reach better performances:

1. Detection $\Longleftrightarrow$ Tracking
2. Tracking $\Longleftrightarrow$ Mapping
3. Classification $\Longleftrightarrow$ Mapping

Consequently we propose in this chapter a grid-based perception algorithm where these three interactions are elegantly integrated. The precise computations performed to solve each perceptual task are successively detailed in the next sections.

We also present experimental results based both on simulated data to valid the key functionalities of the system and on real data to prove the benefit of this approach in the difficult situations presented in Chapter 4.

6.2 Principles

In this section, a brief overview of the three main types of simplifications used in this chapter to implement the proposed pedestrian perception system is given.

6.2.1 Known Localization

Because solving the Localization problem using a perceptual approach did not appear as a critical requirement in our study of pedestrian perception systems, we assume in this chapter that the localization of the sensing platform is known. In our experiments, this localization is provided by the proprioceptive sensors of the sensing platform.

6.2.2 Occupied Cells Sets

The grid-based approach presented in chapter 5 is based on the computation of six probability mass functions in every cell of the environment E (provided that $P(M_k(x_i) = 1|Z_{0:k}) \neq 0$ for all *pmf* except the Mapping one). In practice, most of these *pmf* are only relevant in cells that have a high probability to be occupied. Indeed, computing the velocity estimates of cells that are unlikely to be occupied is feasible but slows down the computations in practice.

It is thus possible to alleviate the computational burden of the whole process by only computing the perceptual *pmfs* on the smaller set of cells that are likely to be occupied. This set noted $\mathcal{OCC}_k$ is computed at time k as follows:

$$\mathcal{OCC}_k = \{x_i \in E / P(M_k(x_i) = 1|Z_{0:k}) \geq \delta\} \qquad \forall \delta \in [0, 1] \tag{6.1}$$

It is interesting to note that by modifying the value chosen for δ, the set $\mathcal{OCC}_k$ can alternatively be very small or as big as E. $\delta = 0.9$ proved to work well in our experiments.

6.2.3 Most Probable Estimates Interactions

Most of the computational burden implied by the grid-based perception approach is linked to the use of discrete summations for integrating the *pmfs* related to other tasks. The use of *pmf* is relevant in lots of situations as multi-modal uncertainty information can for example only

be transmitted properly in this way. However, the most probable value of a given *pmf* (*MAP* estimate) can sometimes be sufficient to transmit relevant information to another task.

As a result, *maximum a posteriori* (*MAP*) estimates will regularly be used instead of marginalizing (summing) over the whole corresponding *pmf*. These *MAP* estimates are indicated by a hat accent as shown below (A_k and B_k can refer to any random variable introduced in Chapter 5).

$$\begin{aligned} P(A_k(x_i)|Z_{0:k}) &= \underbrace{\sum_{b\in\mathcal{B}} P(A_k(x_i)|B_k(x_i)=b, Z_{0:k})P(B_k(x_i)=b|Z_{0:k})}_{\text{Information integrated as a } pmf} \\ &\simeq P(A_k(x_i)|B_k(x_i)=\widehat{b}, Z_{0:k}) \underbrace{\sum_{b\in\mathcal{B}} P(B_k(x_i)=b|Z_{0:k})}_{=1} \\ &\simeq \underbrace{P(A_k(x_i)|B_k(x_i)=\widehat{b}, Z_{0:k})}_{\text{Information integrated as a } MAP \text{ estimate}} \end{aligned} \tag{6.2}$$

Finally, interactions that are not critical for pedestrian perception will simply not be implemented at all.

6.2.4 Outlines of this chapter

The next sections present successively how these simplifications are used to compute successively each perceptual task. For each task, the precise computations performed to solve the problem both locally and globally are detailed and results based on simulated data are shown to present the key characteristic of the system. Finally the last sections of this chapter are dedicated to the analysis on real data of the benefits that such a system can bring to pedestrian perception in highly changing environment.

6.3 Computation of the Association problem

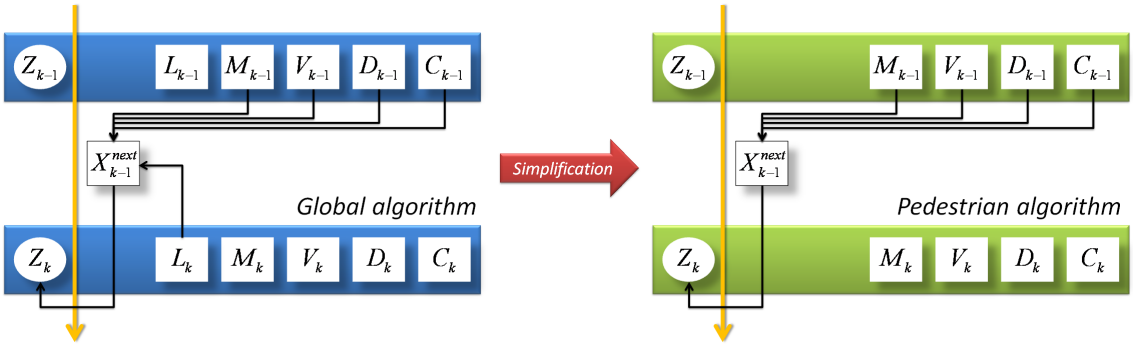


Figure 6.1: Schematic view of the simplifications made in the computation of the Association problem.

6.3.1 Local computation

The local computation of the Association *pmf* is performed on the set $\mathcal{OCC}_{k-1}$ as detailed below:

$\forall x_i \in \mathcal{OCC}_{k-1}$,

$$P_{\text{local}}(X^{next}_{k-1}(x_j)|M_{k-1}(x_j)=1,Z_{0:k}) \simeq \tag{6.3}$$
$$\eta \underbrace{P(z_k|X^{next}_{k-1}(x_j),\widehat{l_k},M_{k-1}(x_j)=1)}_{\text{Sensor Occupancy model}}$$
$$\sum_{v_j\in V}\sum_{c_j\in C} \underbrace{P(X^{next}_{k-1}(x_j)|C_{k-1}(x_j)=c_j,V_{k-1}(x_j)=v_j,M_{k-1}(x_j)=1)}_{\text{Motion model}}$$
$$\underbrace{P(C_{k-1}(x_j)=c_j|M_{k-1}(x_j)=1,Z_{0:k-1})}_{\text{Classification } pmf}$$
$$\underbrace{P(V_{k-1}(x_j)=v_j|M_{k-1}(x_j)=1,Z_{0:k-1})}_{\text{Velocity Estimation } pmf}$$

The sensor occupancy model is similar to the one used in (Elfes, 1989b) and the motion model is a constant velocity model.

6.3.2 Global computation

The global computation of the association process should normally be performed as proposed in chapter 5:

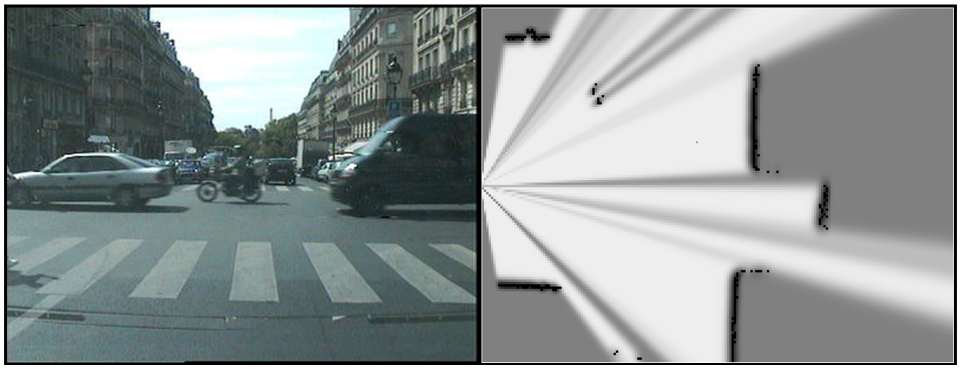


Figure 6.2: Example of a sensor occupancy model (right picture) extracted from the ladar measurements.

$\forall x_i \in \mathcal{OCC}_{k-1}$,

$$P(X_{k-1}^{next}(x_i) = \hat{x} | M_{k-1}(x_i) = 1, Z_{0:k}) \simeq$$
$$\sum_{x_{k-1}^{next} \in \mathcal{A}(\mathcal{OCC}_{k-1})} \left\{ \left\{ \prod_{1 \leq j \leq N} P_{\text{local}}(X_{k-1}^{next}(x_j) | M_{k-1}(x_i) = 1, Z_{0:k}) \right\} \right.$$
$$\left. \Phi_{\text{association}}(x_{k-1}^{next}, \mathcal{OCC}_{k-1}, Z_{0:k}) \right. \quad (6.4)$$

, where $\mathcal{A}(\mathcal{OCC}_{k-1})$ is the set of all the possible global realisations of the reduced random vector $\{X_{k-1}^{next}(y)\}_{y \in \mathcal{OCC}_{k-1}}$ such that $X_{k-1}^{next}(x_i) = \hat{x}$.

Unfortunately the size of $\mathcal{A}(\mathcal{OCC}_{k-1})$ makes this process very computationally demanding in practice. To make the computation of the global Association *pmf* faster, we use the following observations.

1. All the cells contained in the set $\mathcal{OCC}_{k-1}$ belong to a small number of objects. As a result the Association *pmf* of all the cells of an object can be deduced from the Association *pmf* of a few number of specific cells that are called "key cells" and grouped in a set noted $\mathcal{KC}_{k-1}$.

2. The set $\mathcal{A}(\mathcal{OCC}_{k-1})$ of all the possible global association realisations contains some realisations that are very unlikely in practice. For example, investigating a realisation where all the points belonging to a same given object converge to the same cell is useless (its probability of realisation is null). This set can then be reduced to contain only the configurations that are most likely. The corresponding reduced set is noted $\mathcal{B}(\mathcal{OCC}_{k-1})$.

Based on these observations, the computation of the global Association *pmfs* is performed in two steps. First, only "key cells" are associated. Then, all the remaining cells are handled based on the "key cells" Association *pmfs* obtained before. This can be mathematically described as follows:

Step 1 - Global Association of "key cells"

$\forall x_i \in \mathcal{KC}_{k-1}$,

$$P(X_{k-1}^{next}(x_i) = \widehat{x} | M_{k-1}(x_i) = 1, Z_{0:k}) \simeq \sum_{x_{k-1}^{next} \in \mathcal{B}(\mathcal{OCC}_{k-1})} \left\{ \prod_{x_j \in \mathcal{OCC}_{k-1}} P_{\text{local}}(X_{k-1}^{next}(x_j) = x_{k-1}^{next}(x_j) | M_{k-1}(x_j) = 1, Z_{0:k}) \right\} \times \Phi_{\text{Association}}(x_{k-1}^{next}, \mathcal{OCC}_{k-1}, Z_{0:k}) \quad (6.5)$$

Step 2 - Global Association of the remaining cells

$\forall x_i \in \mathcal{OCC}_{k-1}/\mathcal{KC}_{k-1}$,

$$P(X_{k-1}^{next}(x_i) = \widehat{x} | M_{k-1}(x_i) = 1, Z_{0:k}) \simeq \sum_{x_{k-1}^{next} \in \mathcal{B}(\mathcal{KC}_{k-1})} \left\{ \prod_{x_j \in \mathcal{KC}_{k-1}} \underbrace{P(X_{k-1}^{next}(x_j) = x_{k-1}^{next}(x_j) | M_{k-1}(x_j) = 1, Z_{0:k})}_{\text{Computed in step 1}} \right\} \times \Phi_{\text{Association}}(x_{k-1}^{next}, \mathcal{KC}_{k-1}, Z_{0:k}) \quad (6.6)$$

Key Cells Selection

To select "key cells", the object-based representation introduced in the previous chapter is used to decide which cells belong to the same objects in the environment based on the information available in the Detection *pmf*.

Objects are first grouped in a set noted $\mathcal{OBJ}_{k-1}^{\delta}$:

$$\mathcal{OBJ}_{k-1}^{\delta} = \left\{ \mathbb{O}_{k-1}^{\delta}(x_i) \right\}_{x_i \in \mathcal{OCC}_{k-1}}$$

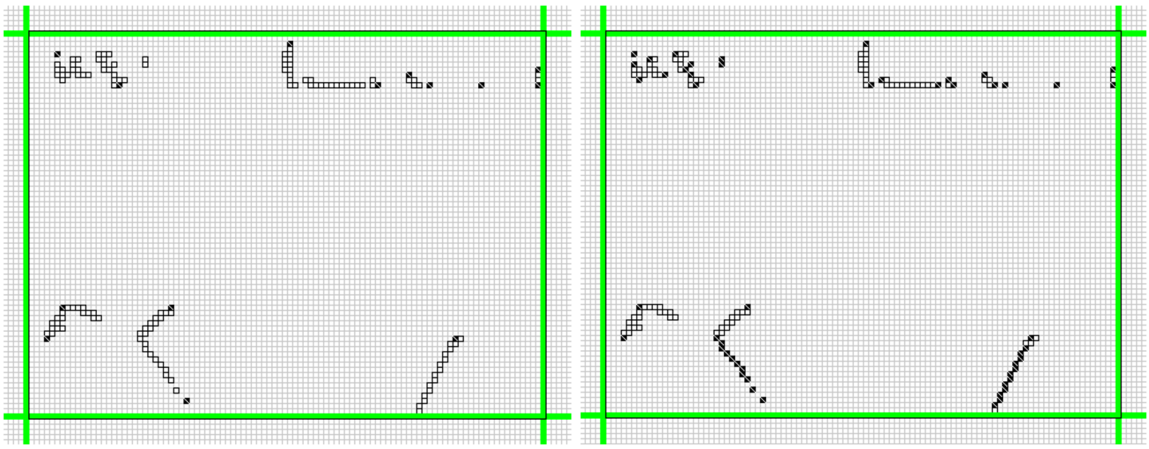
, where

$$\mathbb{O}_{k-1}^{\delta}(x_i) = \{ x_j \in E \setminus P(D_{k-1}(x_i, x_j) = 1 | M_{k-1}(x_i, x_j) = 1, Z_{0:k-1}) \geq \delta \}$$

Finally, for each object $obj \in \mathcal{OBJ}_{k-1}^{\delta}$, a maximum of two "key cells" $x_a \in obj$ and $x_b \in obj$ are selected such that:

$$(x_a, x_b) = \arg \max_{(x_i, x_j) \in obj^2} |x_i - x_j|$$

It is important to note that depending on the value chosen for δ, the size of the set $\mathcal{KC}_{k-1}$ can vary a lot (equal to $\mathcal{OCC}_{k-1}$ if $\delta = 0$).



(a) $\delta = 0.9$ (b) $\delta = 0.4$

Figure 6.3: Example of Key Cells selected with different values for δ.

Note that selecting very few "key cells" will ensure fast computations but amounts performing anticipated and potentially erroneous object detection. By selecting a fair amount of "key cells", we ensure that potential errors on the estimated *pmf* of D_{k-1} will not undermine the association procedure.

Feasible Realisation Selection

Because the sets $\mathcal{B}(\mathcal{OCC}_{k-1})$ and $\mathcal{B}(\mathcal{KC}_{k-1})$ are intended to only contain the association realisation that are globally feasible, these sets are selected using the potential function $\Phi_{\text{Association}}$ as follow:

$$\mathcal{B}(\mathcal{OCC}_{k-1}) = \{a \in \mathcal{A}(\mathcal{OCC}_{k-1}) / \Phi_{\text{Association}}(a, \mathcal{OCC}_{k-1}, Z_{0:k}) \geq \gamma \}$$

$$\mathcal{B}(\mathcal{KC}_{k-1}) = \{a \in \mathcal{A}(\mathcal{KC}_{k-1}) / \Phi_{\text{Association}}(a, \mathcal{KC}_{k-1}, Z_{0:k}) \geq \gamma \}$$

It is also important to mention that the computational burden implied by the computation of the Association problem can vary a lot depending on the value given to γ. Note for example that a heavy but complete computation over all the possible realisations can be obtained by setting $\gamma = 0$.

Potential Function $\Phi_{\text{Association}}$

The function $\Phi_{\text{Association}}(E, a, Z_{0:k})$ is implemented in such a way that the two following situations are penalized.

- In the association realisation a, the distance between two points belonging to the same object undergoes a severe distortion.
- In the association realisation a, two points belonging to different objects are converging to the same cell.

To implement these two simple heuristics, the information contained in the Detection *pmf* is required. This is done as follows.

$$\Phi_{\text{Association}}(x_{k-1}^{next}, H, Z_{0:k}) = \prod_{(x,y)\in H} \alpha(x_{k-1}^{next}, x, y) P(D_{k-1}(x,y) = 1 | M_{k-1}(x,y) = 1, Z_{0:k-1}) \\ + \beta(x_{k-1}^{next}, x, y) P(D_{k-1}(x,y) = 0 | M_{k-1}(x,y) = 1, Z_{0:k-1}) \tag{6.7}$$

, where the terms α and β are chosen such that the two heuristics mentioned above are implemented. In our experiments we used:

$$\alpha(x_{k-1}^{next}, x, y) = 1 - \rho \left| \frac{\left| x^{next}(x) - x^{next}(y) \right|}{|x - y|} - 1 \right|$$

$$\beta(x_{k-1}^{next}, x, y) = (1 - \epsilon) \mathbb{1}_{x^{next}(x) \neq x^{next}(y)} + \epsilon \mathbb{1}_{x^{next}(x) = x^{next}(y)}$$

, where ρ and ϵ are parameters chosen appropriately.

6.3.3 Experimental validation

The association method described in this section allows to successfully associate the former occupancy map with the new measurements as shown on simulated data in figure 6.4. Experiments conducted on real data will be presented in the last sections of this chapter.

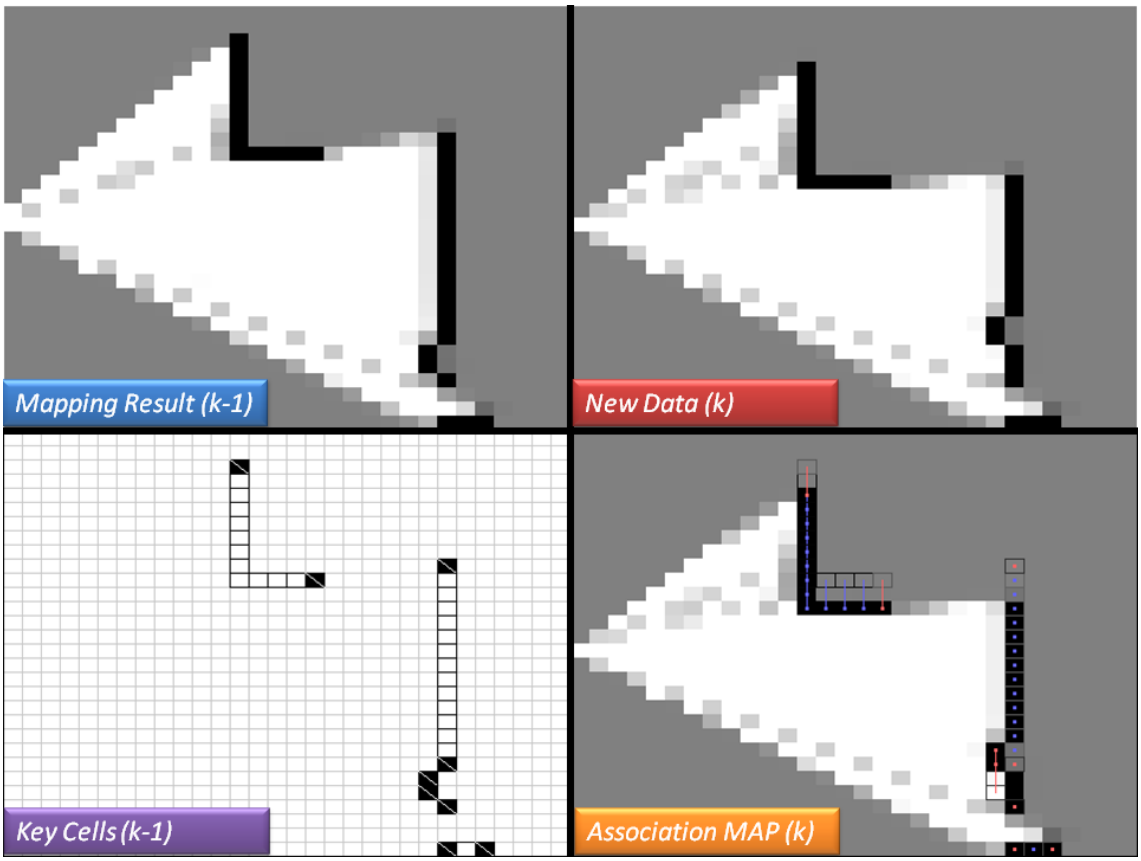


Figure 6.4: Association example on simulated data. The sensing vehicle is on the left of each top-view pictures. The top-left picture shows the occupancy map at time $k-1$. The top-right picture shows the new incoming ladar data at time k. The bottom-left pictures shows the "key cells" detected for association and the bottom-right picture show the most probable association solution (maximum a posteriori).

6.4 Computation of the Mapping problem

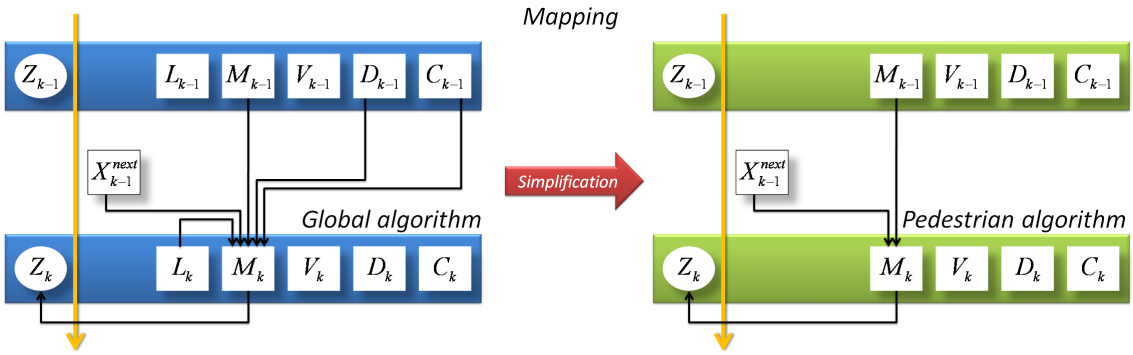


Figure 6.5: Schematic view of the simplification made in the computation of the Mapping problem.

6.4.1 Local computation

The local computation of the Mapping *pmf* is performed on every environment cell as detailed below:

$\forall x_i \in E$,

$$
\begin{aligned}
&P_{\text{local}}(M_k(x_i) = 1 | Z_{0:k}) = \\
&\quad \underbrace{P_{\text{local}}(M_k(x_i) = 1 | S_k(x_i) = 1, Z_{0:k})}_{\text{Mapping if already seen}} \ \underbrace{P(S_k(x_i) = 1 | Z_{0:k})}_{\text{Status of the cell}} \\
&\quad + \underbrace{P_{\text{local}}(M_k(x_i) = 1 | S_k(x_i) = 0, Z_{0:k})}_{\text{Mapping if never seen}} \ P(S_k(x_i) = 0 | Z_{0:k})
\end{aligned}
\tag{6.8}
$$

Status of the cell

As already mentioned in chapter 5, the following method is used to estimate the probability for a cell to be seen for the first time.

$$
\begin{aligned}
&P(S_k(x_i) = 1 | Z_{0:k}) = \\
&\quad \begin{cases} \gamma & \text{if } \exists x_j \in E \text{ st } x_i = \arg\max_{x_l} P(X_{k-1}^{next}(x_j) = x_l | M_{k-1}(x_j) = 1, Z_{0:k}) \\ 1 - \gamma & \text{else} \end{cases}
\end{aligned}
\tag{6.9}
$$

Mapping if already seen

In this computation, the information contained in the previous Mapping *pmf* is used to update the new occupancy of each cell. This is naturally performed by summing over all the

association hypothesis with former cells (weighted by the Association *pmf*). No interaction with the Classification task is implemented here.

$$P_{\text{local}}(M_k(x_i) = 1 | S_k(x_i) = 1, Z_{0:k}) = \tag{6.10}$$
$$\beta\eta \sum_{x_j \in E} \underbrace{P(z_k | M_k(x_i) = 1, X_{k-1}^{next}(x_j) = x_i, M_{k-1}(x_j) = 1, \widehat{l_j})}_{\text{Sensor Free Space model}}$$
$$\underbrace{P(M_k(x_i) = 1 | M_{k-1}(x_j) = 1, X_{k-1}^{next}(x_j) = x_i)}_{\text{Occupancy Evolution model}}$$
$$\underbrace{P(M_{k-1}(x_j) = 1 | Z_{0:k-1})}_{\text{Mapping } pmf}$$
$$\underbrace{P(X_{k-1}^{next}(x_j) = x_i | M_{k-1}(x_j) = 1, Z_{0:k})}_{\text{Association } pmf}$$

, where $\beta = (\sum_{x_j \in E} P(X_{k-1}^{next}(x_j) = x_i | M_{k-1}(x_j) = 1, Z_{0:k}))^{-1}$.

Two models need to be defined here. The first is the evolution model of the occupancy of one cell given the fact that it was occupied one step before. We chose to estimate this model as a simplistic constant value μ. This value is important as it parameterizes the ability of the system to maintain in the map object points that are not visible anymore. A high value of μ will lead to highly detailed map (aggregating all the past measurements) while a small value of this parameter will make the estimated objects outlines past away progressively.

The sensor free space model is derived from the common sensor occupancy model. It differs however from the sensor occupancy model by only penalizing free spaces in the environment. Occupied cells and unknown cells are given the same high probability. Using this specific sensor model is necessary to allow object points that have been seen before to be maintained in areas where they are not visible anymore. A representation of this model is given in picture 6.6.

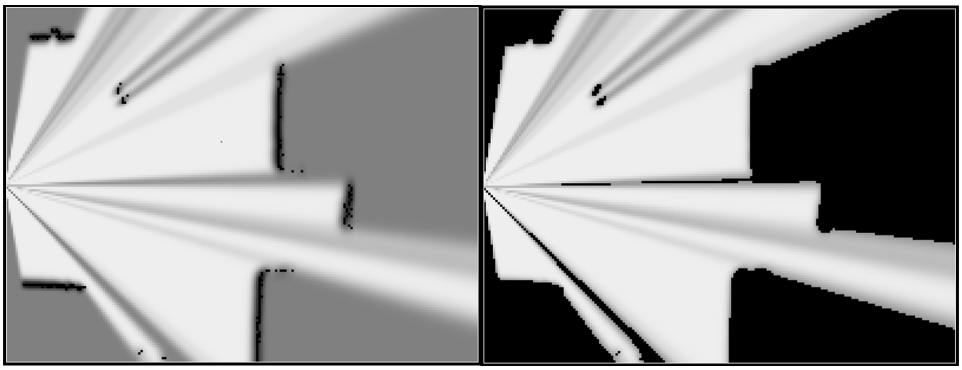


Figure 6.6: The sensor free space model (right) compared to the sensor occupancy model (left).

Mapping if never seen

If the cell is seen for the first time, its probability of occupancy is directly derived from the sensor observation weighted by the prior probability of a cell to be occupied .

$$P_{\text{local}}(M_k(x_i) = 1 | S_k(x_i) = 0, Z_{0:k}) = \eta \underbrace{P(z_k | M_k(x_i) = 1, \widehat{l_j})}_{\text{Sensor Occupancy model}} \underbrace{P(M_k(x_i) = 1 | S_k(x_i) = 0, Z_{0:k-1})}_{\text{Probability of Apparition}} \tag{6.11}$$

The prior probability of occupancy can be derived from the proximity of that cell with the edges of the sensor field of view for example. In our experiment, a simple constant value ν was chosen.

6.4.2 Global computation

It is assumed here that a correct Mapping can be achieved without taking into account interactions with the others cells. Hence,
$\forall x_i \in E$,

$$P(M_k(x_i) = 1 | Z_{0:k}) \simeq P_{\text{local}}(M_k(x_i) = 1 | Z_{0:k}) \tag{6.12}$$

This assumption proved to be valid in practice as most of the relevant information about interactions between neighbouring cells is already embedded in the Association *pmf* used in the Mapping computation.

6.4.3 Experimental results

Figure 6.7 shows an example of a scene where a vehicle is moving from the left of the sensing vehicle to the right and a pedestrian is moving in the opposite direction along a wall. In this example, the proposed algorithm is able to build a relevant map of the changing environment.

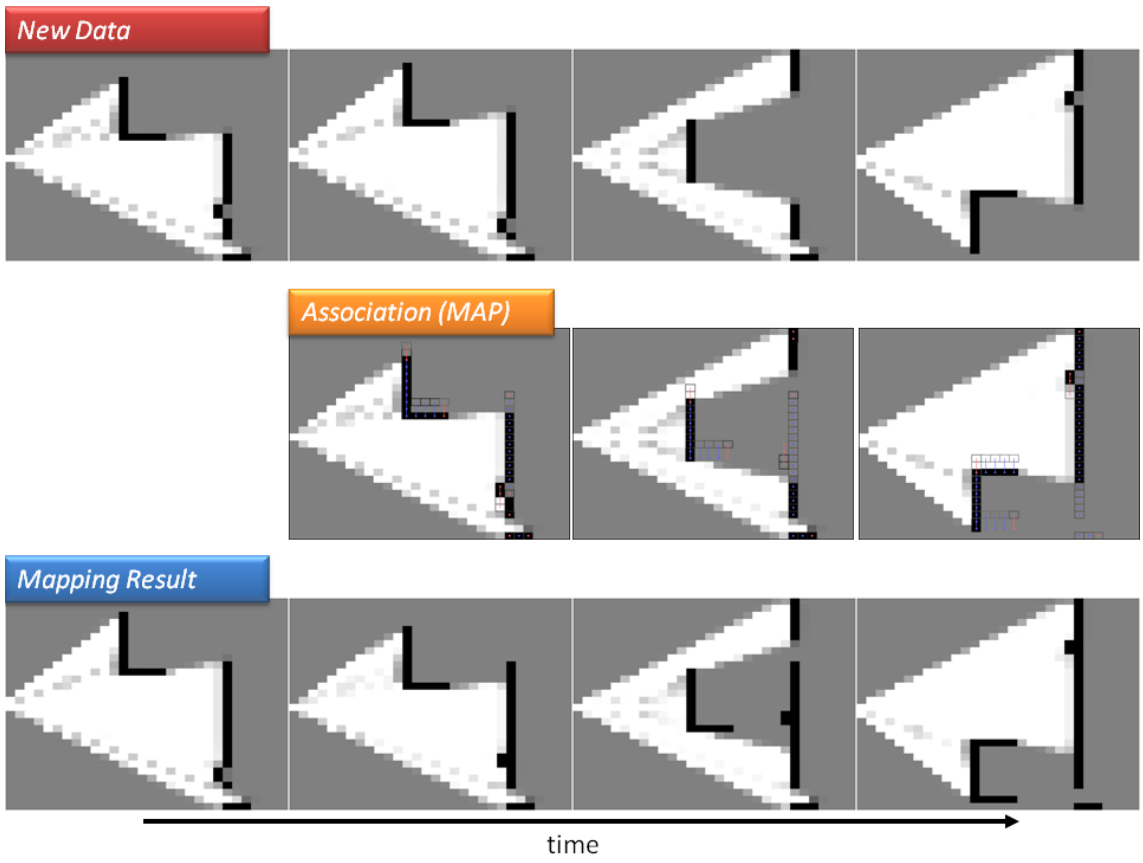


Figure 6.7: Mapping example on simulated data. The sensing vehicle is on the left of each picture. The successive occupancy maps show the progressive estimation of the tracked vehicle outlines. Note that the outlines of the "pedestrian" are also maintained even when they are temporarily occluded.

6.5 Computation of the Velocity Estimation problem

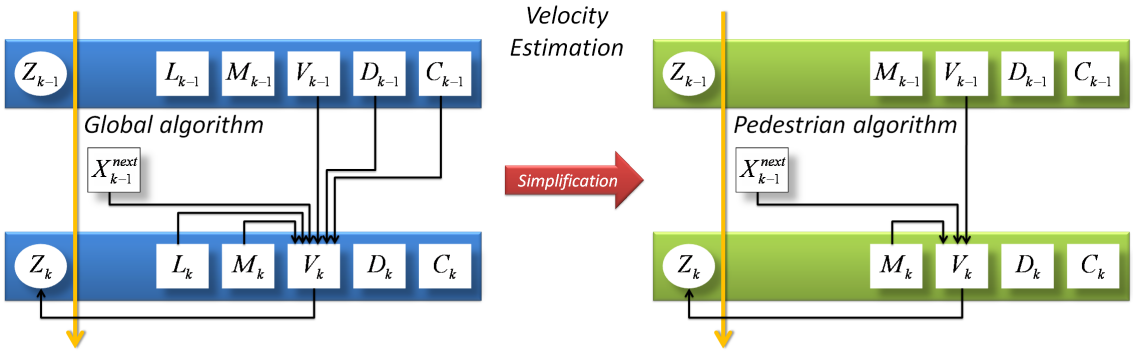


Figure 6.8: Schematic view of the simplification made in the computation of the Velocity Estimation problem.

6.5.1 Local computation

The local computation of the Velocity Estimation problem is performed using the equations below. Note that this computation is only performed on cells that are likely to be occupied ($\mathcal{OCC}_k$).

$\forall x_i \in \mathcal{OCC}_k$,

$$P_{\text{local}}(V_k(x_i)|M_k(x_i)=1,Z_{0:k}) = \tag{6.13}$$

$$\underbrace{P_{\text{local}}(V_k(x_i)|S_k(x_i)=1,M_k(x_i)=1,Z_{0:k})}_{\text{Velocity Estimation if already seen}} \quad \underbrace{P(S_k(x_i)=1|Z_{0:k})}_{\text{Status of the cell}} \tag{6.14}$$

$$+\underbrace{P_{\text{local}}(V_k(x_i)|S_k(x_i)=0,M_k(x_i)=1,Z_{0:k})}_{\text{Velocity Estimation if never seen}} \quad P(S_k(x_i)=0|Z_{0:k}) \tag{6.15}$$

Status of the cell

This probability is computed as shown in the previous sections.

Velocity Estimation if already seen

If the point occupying a cell x_i has already been seen before, the estimation of its new velocity (acceleration, etc...) is based on its previous velocity and on the information given by the Association *pmf* about its last displacement.

$$P_{\text{local}}(V_k(x_i)|S_k(x_i)=1,M_k(x_i)=1,Z_{0:k}) = \tag{6.16}$$
$$\beta \sum_{x_j \in E} \sum_{v_j \in V} \underbrace{P(V_k(x_i)|V_{k-1}(x_j)=v_j, X_{k-1}^{next}(x_j)=x_i, M_k(x_i)=1, M_{k-1}(x_i)=1)}_{\text{Velocity evolution model}}$$
$$\times \underbrace{P(V_{k-1}(x_j)=v_j|M_{k-1}(x_j)=1,Z_{0:k-1})}_{\text{Velocity Estimation } pmf}$$
$$\times \underbrace{P(X_{k-1}^{next}(x_j)=x_i|M_{k-1}(x_j)=1,Z_{0:k})}_{\text{Association } pmf}$$

The velocity evolution model is based on a Gaussian:

$$P(V_k(x_i)=v_i|V_{k-1}(x_j)=v_j, X_{k-1}^{next}(x_j)=x_i, M_k(x_i)=1, M_{k-1}(x_i)=1) = \mathcal{N}(v_i,\mu,\Sigma)$$

$$\mu = \frac{x_i - x_j}{\Delta t} \qquad \Sigma = \begin{pmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{pmatrix}$$

Velocity if never seen

When the point occupying a cell x_i has never been seen before, a non informative prior is used:

$$P(V_k(x_i)|S_k(x_i)=0, M_k(x_i)=1, Z_{0:k}) = \frac{1}{\text{card}(V)} \tag{6.17}$$

6.5.2 Global computation

It is assumed here that a satisfying Velocity Estimation can be obtained without taking into account the other cells directly. Hence:

$\forall x_i \in \mathcal{OCC}_k$,

$$P(V_k(x_i)|M_k(x_i)=1,Z_{0:k}) \simeq P_{\text{local}}(V_k(x_i)|M_k(x_i)=1,Z_{0:k}) \tag{6.18}$$

6.5.3 Experimental results

Figure 6.9 shows an example of the Velocity estimations performed on simulated ladar data in a specific situation. It is interesting to note that the algorithm is in this case able to correctly estimate the objects velocities. A point-based algorithm would certainly fail due to the progressive occlusion of the "wall".

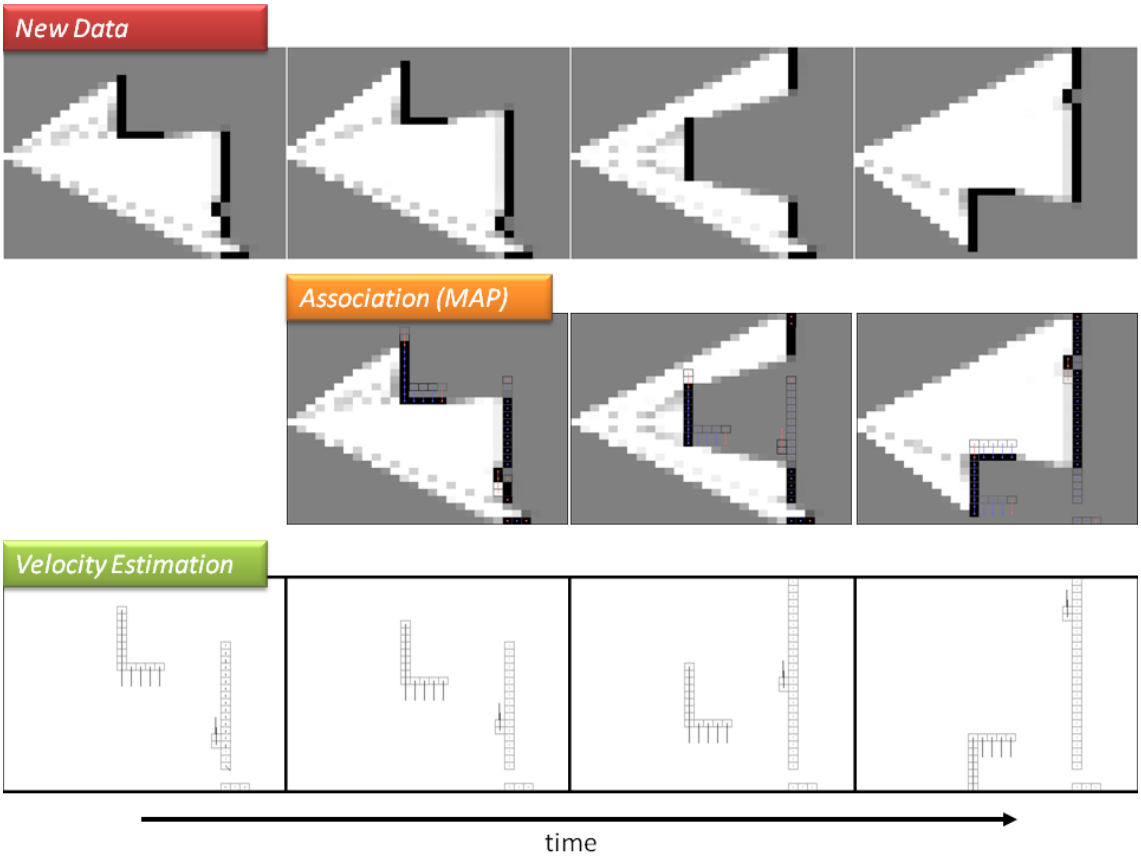


Figure 6.9: Tracking example on simulated data. The two cells moving closely to the wall are successfully tracked.

6.6 Computation of the Detection problem

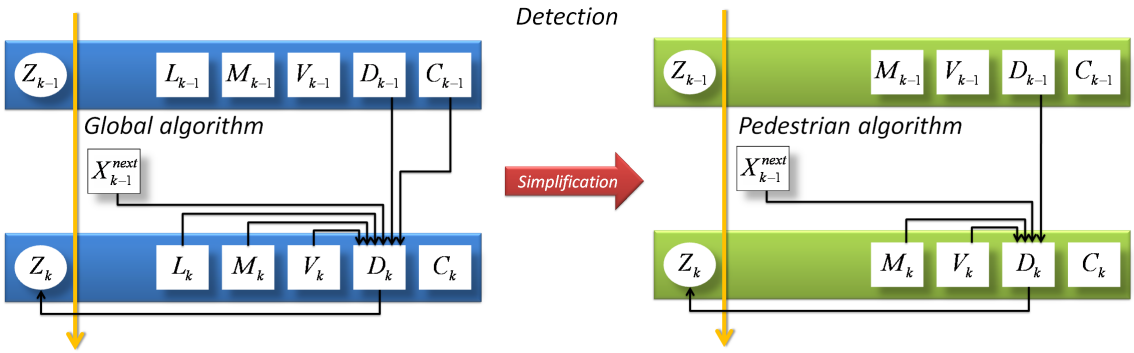


Figure 6.10: Schematic view of the simplification made in the computation of the Detection problem.

6.6.1 Local computation

The local computation of the Detection *pmf* is performed based on the geometric information contained in the Mapping *pmf* and on the dynamic information estimated through the computation of the Velocity Estimation *pmf*. This can be described by the following equations:

$\forall (x_i, x_j) \in (\mathcal{OCC}_k)^2$,

$$\begin{aligned} P_{\text{local}}(D_k(x_i,x_j)=1|M_k(x_i,x_j)=1,Z_{0:k}) = & \\ \underbrace{P_{\text{local}}(D_k(x_i,x_j)=1|S_k(x_i,x_j)=1,M_k(x_i,x_j)=1,Z_{0:k})}_{\text{Detection if already seen}} & \underbrace{P(S_k(x_i,x_j)=1|Z_{0:k})}_{\text{Status of the cells}} \\ + \underbrace{P_{\text{local}}(D_k(x_i,x_j)=1|S_k(x_i,x_j)=0,M_k(x_i,x_j)=1,Z_{0:k})}_{\text{Detection if never seen before}} & P(S_k(x_i,x_j)=0|Z_{0:k}) \end{aligned} \tag{6.19}$$

Status of the cells

This term is computed as follows:

$$\begin{aligned} P(S_k(x_i,x_j)=1|M_k(x_i,x_j)=1) = & \\ P(S_k(x_i)=1|M_k(x_i)=1) \times P(S_k(x_j)=1|M_k(x_j)=1) & \end{aligned} \tag{6.20}$$

, where the terms of the product are computed as discussed in the previous steps of this algorithm.

Detection if already seen

The two cells implied in this computation make the use of whole *pmf* computationally demanding. Therefore, the interaction with the Classification task is not implemented here and the information coming from the Velocity Estimation task is integrated as *a MAP* estimate.

$$P(D_k(x_i,x_j)=1|S_k(x_i,x_j)=1,M_k(x_i,x_j)=1,Z_{0:k}) = \\ P(D_k(x_i,x_j)=1|V_k(x_i)=\widehat{v_i},V_k(x_j)=\widehat{v_j},S_k(x_i,x_j)=1,M_k(x_i,x_j)=1,Z_{0:k}) \quad (6.21)$$

To compute this term, a trapezoidal function g is used as follows:

$$P(D_k(x_i,x_j)=1|V_k(x_i)=\widehat{v_i},V_k(x_j)=\widehat{v_j},S_k(x_i,x_j)=1,M_k(x_i,x_j)=1,Z_{0:k}) = \\ g(|\widehat{v_j}-\widehat{v_i}|)_{(\alpha_1,\alpha_2)} \times g(|x_j-x_i|)_{(\alpha'_1,\alpha'_2)} \quad (6.22)$$

, where

$$g(y)_{(\alpha_1,\alpha_2)} = \begin{cases} 1 & \text{if } 0 \leq y < \alpha_1 \\ \frac{\alpha_2 - y}{\alpha_2 - \alpha_1} & \text{if } \alpha_1 \leq y < \alpha_2 \\ 0 & \text{if } \alpha_2 \leq y \end{cases}$$

Detection if never seen

When one of the two points occupying the cells (x_i,x_j) has never been seen before, the Detection computation is only based on the proximity of the two points:

$$P_{\text{local}}(D_k(x_i,x_j)=1|S_k(x_i,x_j)=1,M_k(x_i,x_j)=1,Z_{0:k}) = g(|x_j-x_i|)_{(\alpha'_1,\alpha'_2)} \quad (6.23)$$

6.6.2 Global computation

As mentioned in the previous chapter, the local computation of the Detection *pmf* is not sufficient to obtain a consistent global Detection *pmf*. Indeed, cells A and B can be estimated as belonging to the same object, as cells B and C, but the probability that A and C belong to the same object can still being equal to zero. To address this problem, a simple deterministic approach is used such that, $\forall (x_A,x_B,x_C) \in (\mathcal{OCC}_k)^3$,

$$P(D(x_A,x_C)|M_k(x_A,x_C)=1,Z_{0:k}) = \\ P(D(x_A,x_B)|M_k(x_A,x_B)=1,Z_{0:k}) \times P(D(x_B,x_C)|M_k(x_B,x_C)=1,Z_{0:k}) \quad (6.24)$$

6.6.3 Experimental results

Figure 6.11 shows a situation where the two moving cells are successfully grouped together and discriminated from the wall. This is a situation that would probably be misleading for algorithms using common approaches.

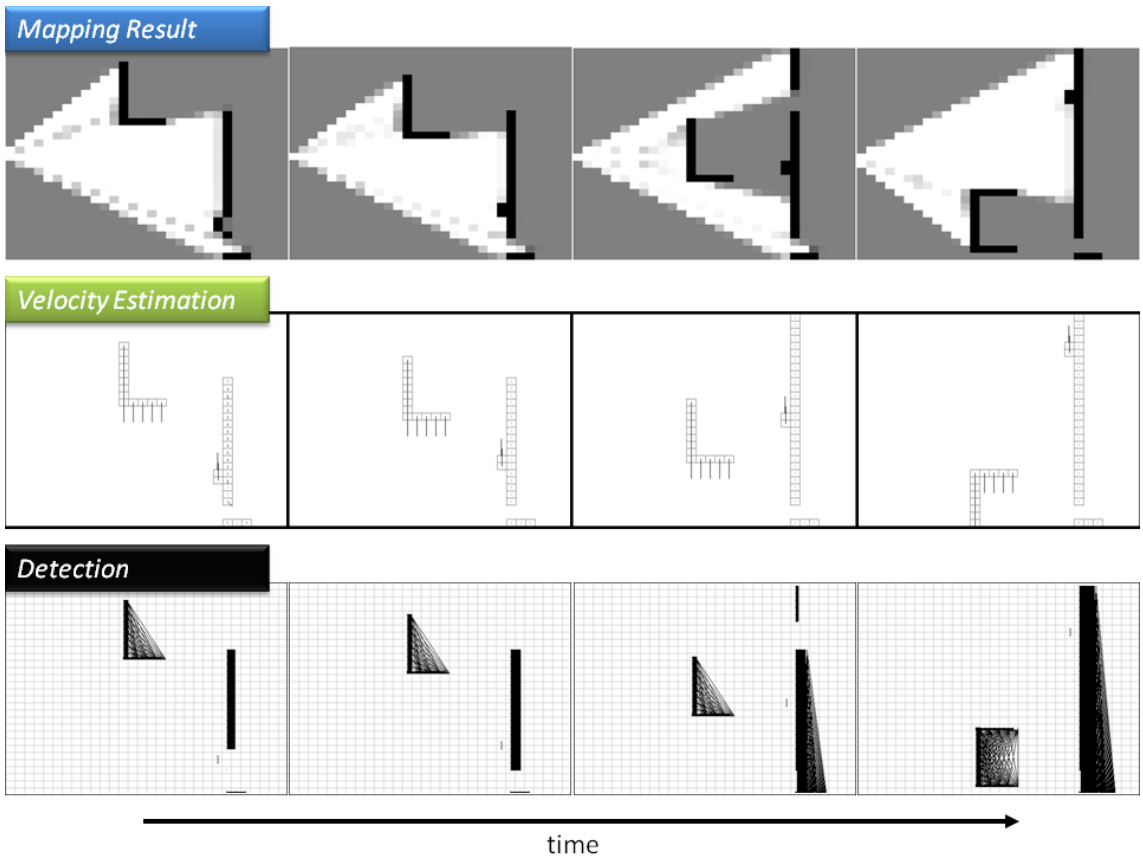


Figure 6.11: Example of a situation where dynamical features are required to solve correctly the Detection problem. Indeed, without the help of the velocity estimation, the two cells representing the pedestrian walking along a wall in the figure above would certainly have been merged with the "wall".

6.7 Computation of the Classification problem

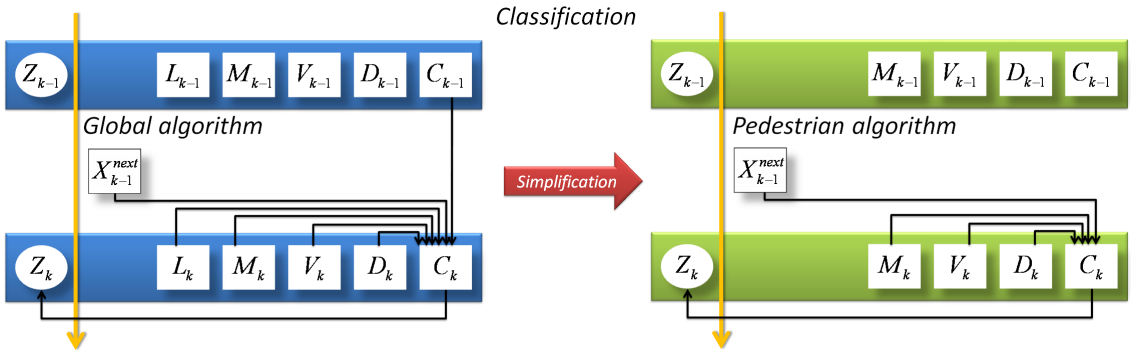


Figure 6.12: Schematic view of the simplification made in the computation of the Classification problem.

6.7.1 Local computation

No local computation of the Classification *pmf* is performed here.

6.7.2 Global computation

For simplicity, the computation of the global Classification *pmf* is only performed through the following equation:

$$P(C_k(x_i)|M_k(x_i)=1,Z_{0:k}) = \underbrace{P(C_k(x_i)|V(\mathbb{O}_\delta(x_i))=v,\mathbb{O}_\delta(x_i) \text{ is real},M_k(x_i)=1,Z_{0:k})}_{\text{Computed by an object-based algorithm}} \tag{6.25}$$

The computation of this term is performed using the ladar-based classification algorithm detailed in chapter 3. Note that the Classification *pmf* of time $k-1$ is not directly used here. While this would certainly lead to inconsistent Classification using common approaches (when objects are temporarily occluded for example), it is not critical in this approach to integrate the former Classification knowledge in this computation. Indeed, the information contained in the Mapping *pmf* is richer than in common approaches (the map of the moving objects are estimated) and is sufficient to ensure reliable Classification in practice.

6.8 Results for Pedestrian Perception in Difficult Situations

This section is intended to valid the original approach proposed in Chapter 5 and 6 by analysing its benefits on real ladar data collected in highly changing environments. These benefits are in particular assessed on the three specific situations described in Chapter 4 where most common approaches fail.

6.8.1 Computationally Demanding Approach

The first experimental result that should be presented is the computational requirements implied by the proposed approach. Despite the simplifications detailed in this chapter to adapt the general approach proposed in Chapter 5, this algorithm is still more computationally demanding than the pedestrian perception system presented in Chapter 3.

First, this algorithm is slower than many existing approaches. In our experiments, processing a ladar scan takes approximatively *800 milliseconds* when the system described in Chapter 3 only needs *12 milliseconds*. This difference is significant and is undoubtedly a critical problem for onboard implementations on nowadays architectures.

Second, storing the probability mass functions related to the five perceptual tasks requires a significant amount of memory. Memory requirements are highly dependent on the resolution and the size of the grid used to model the environment. For reasonable resolutions (cells of 10cm $\times$ 10cm) and with a modelled environment of 20m $\times$ 20m, the system requires $\simeq$500Mo.

This new pedestrian perception system is thus undoubtedly more computationally demanding that the system proposed in Chapter 3. We should mention nevertheless that significant improvements can certainly be achieved on these aspects by using solutions proposed in the literature to alleviate the computational burden implied by grid-based approaches. In particular multi-resolution approaches as proposed in (Kraetzschmar *et al.*, 2004) would certainly allow better computational performances.

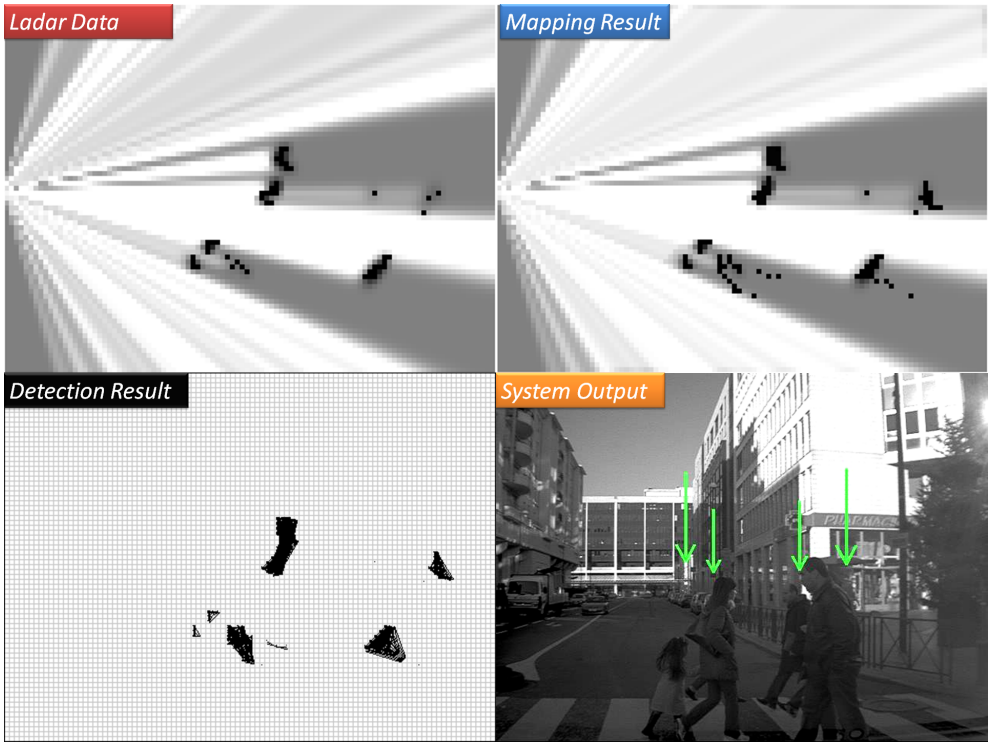
However, while being, in its current form, more demanding in terms of computational power than most existing approaches, the proposed system has uncommon and noticeable perception capabilities that are described in the next paragraphs.

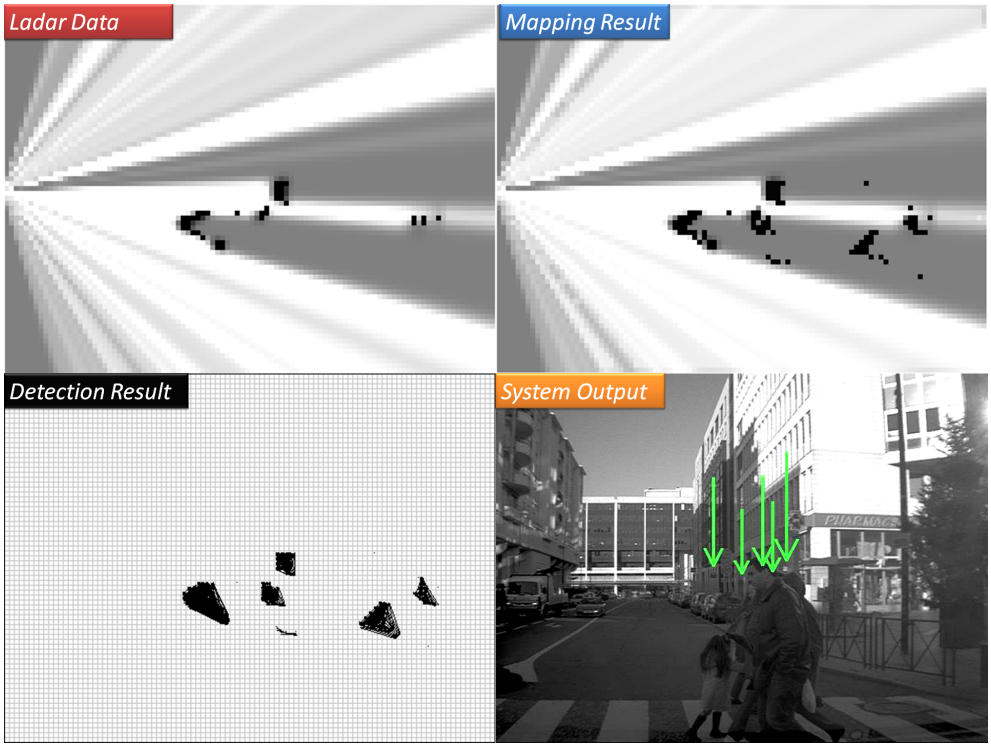
6.8.2 Enhanced Detection

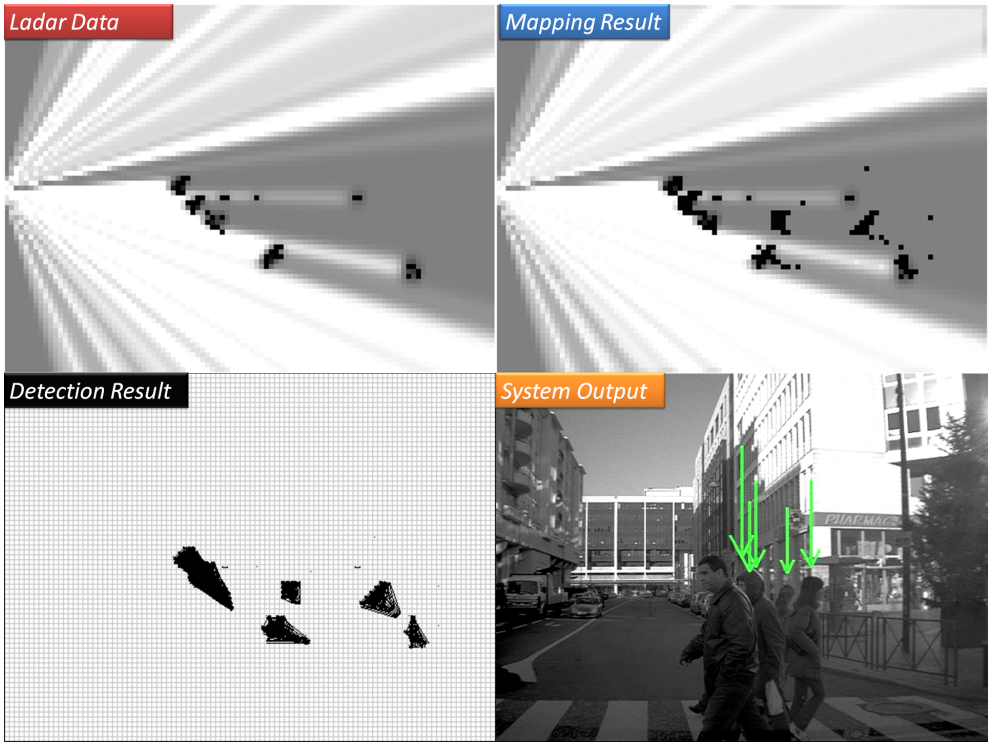
Description

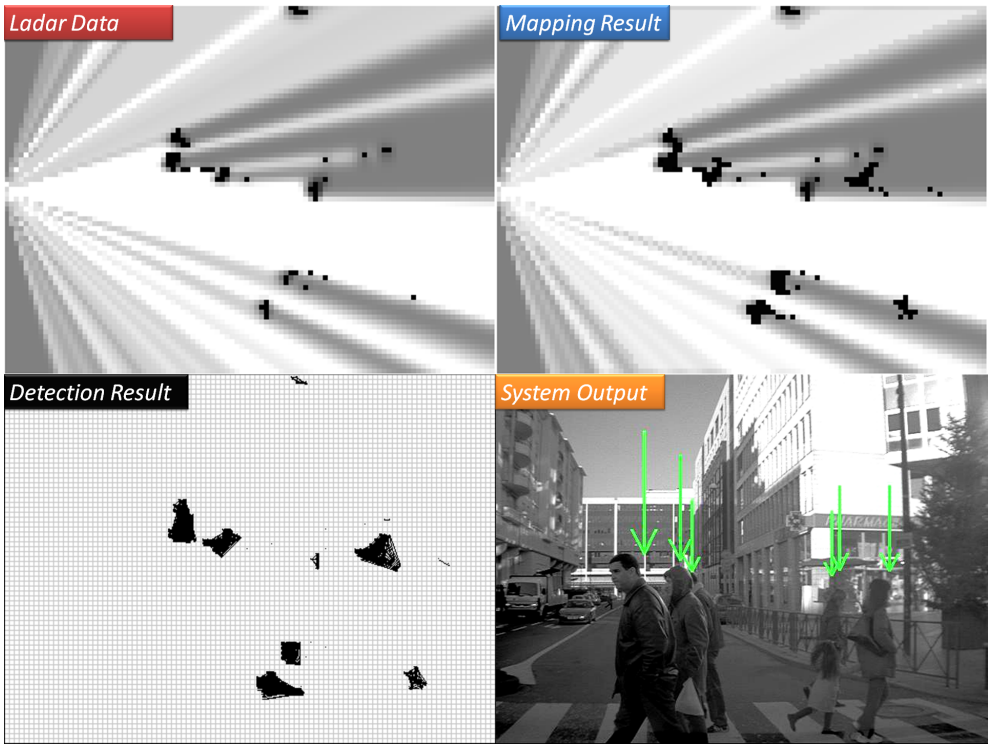
The first type of situations that are very difficult to handle for a perception system are situations where objects are moving very closely from each others and are heavily occluded. As discussed in Chapter 4, the accurate detection of these obstacles is problematic. By enabling the interaction between Detection and Tracking, the algorithm proposed in this chapter is now able to cope with this situation elegantly. Presenting these results is not easy in a written document. An overview of the algorithm behavior in this situation is nevertheless given through successive representations of the system *pmfs* and the visualisation of the system output on a calibrated camera image.

Results - Example 1

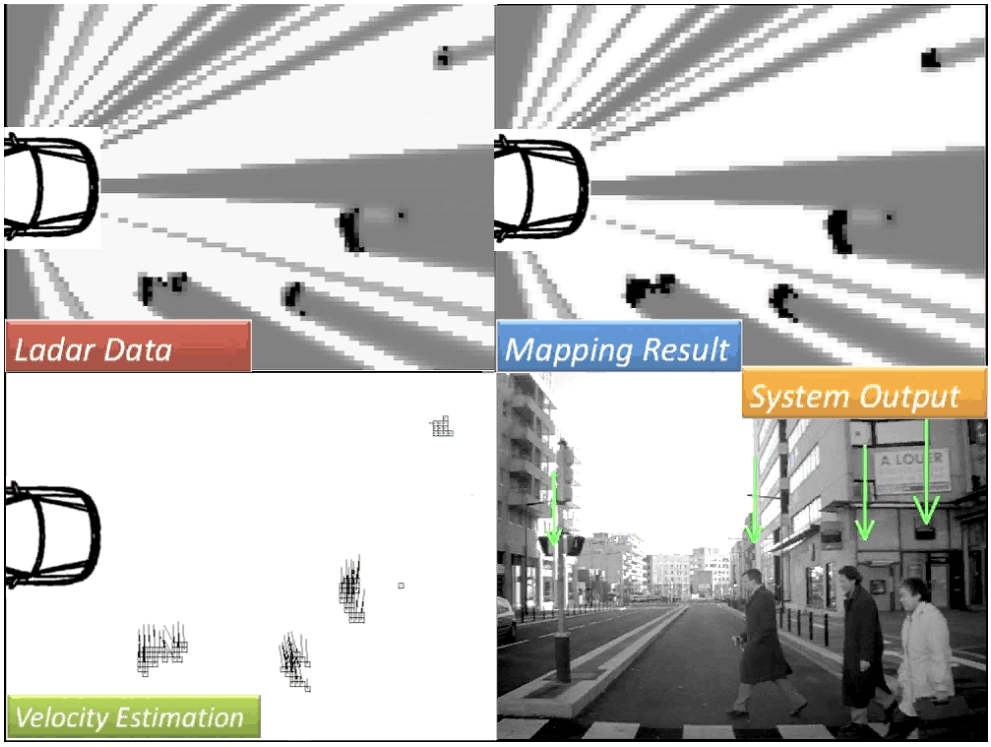


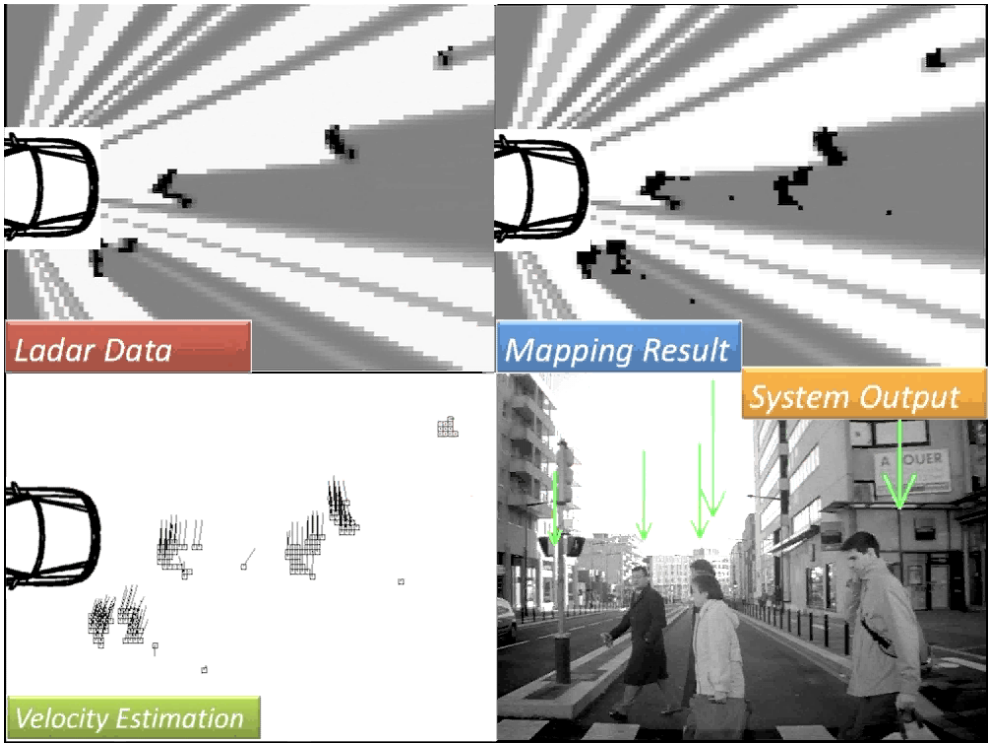


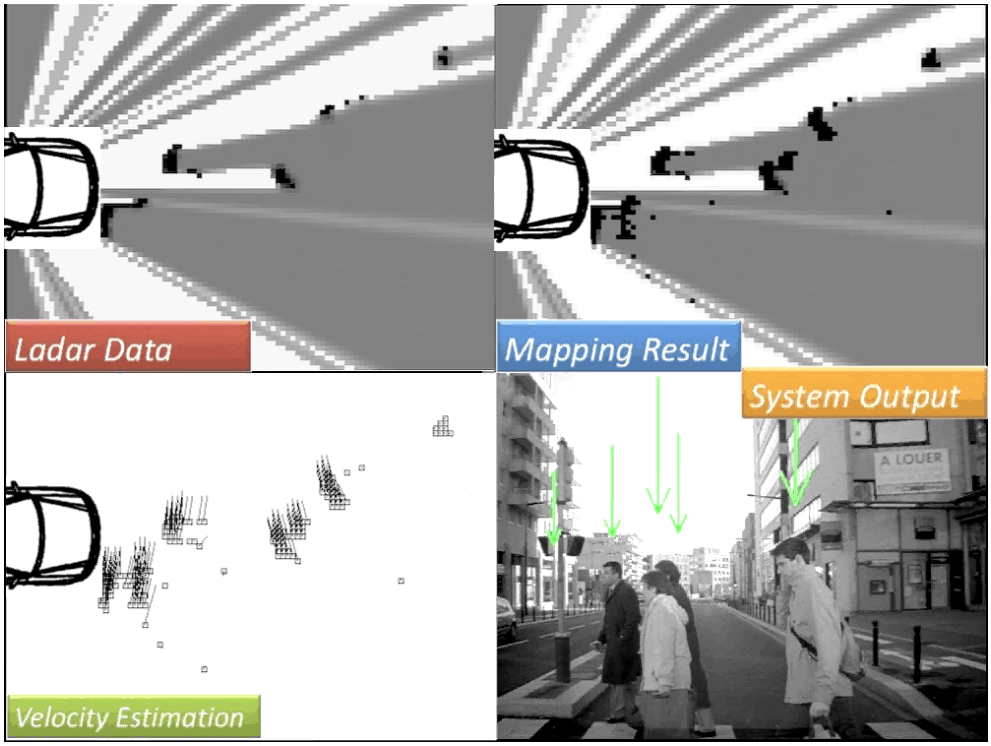

Ladar Data
Mapping Result
Detection Result
System Output

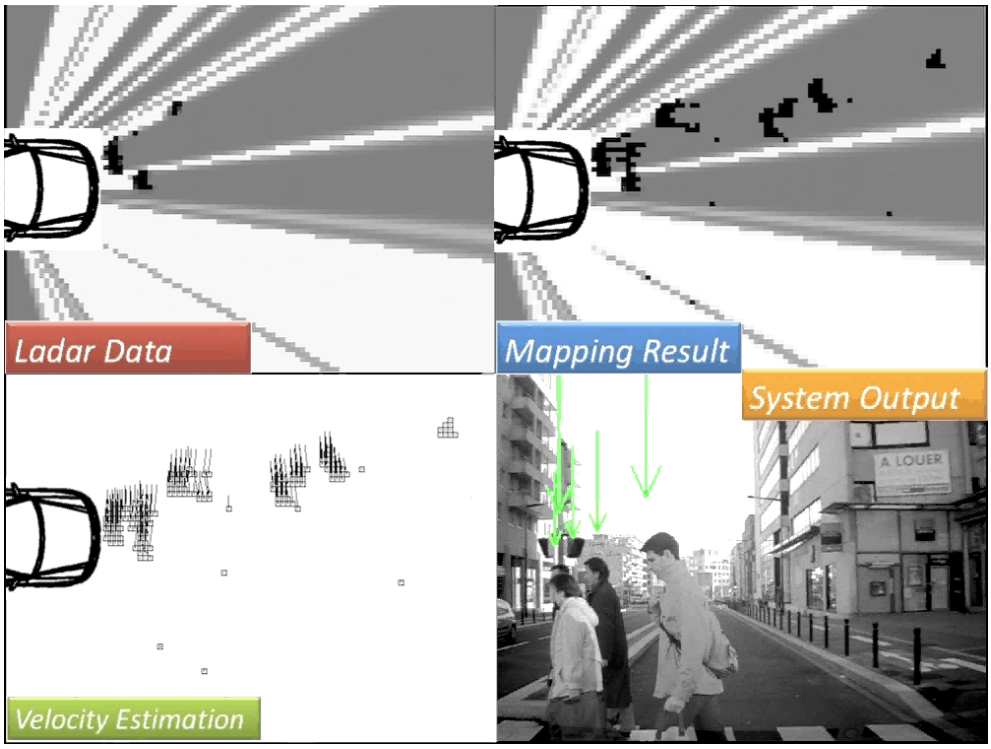

Ladar Data
Mapping Result
Detection Result
System Output

Results - Example 2






Ladar Data
Mapping Result
System Output
Velocity Estimation
A LOUER


Ladar Data
Mapping Result
System Output
Velocity Estimation
A LOUER

Analysis

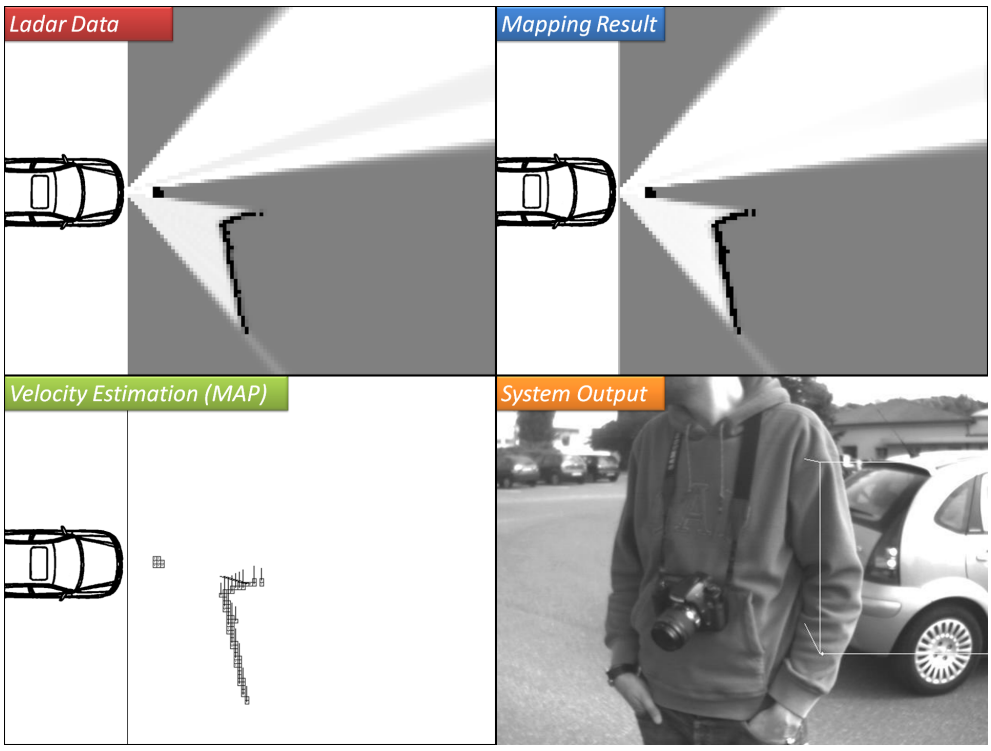
On these two examples all the pedestrians present in this sequence are correctly detected despite the severe occlusions that deteriorate the ladar measurements. This correct detection is achieved thanks to the accurate occupancy map that is iteratively estimated and the velocity estimations that permit to discriminate every object despite their close proximity. Handling groups of pedestrians in this way is a very desirable capability for a pedestrian perception system.

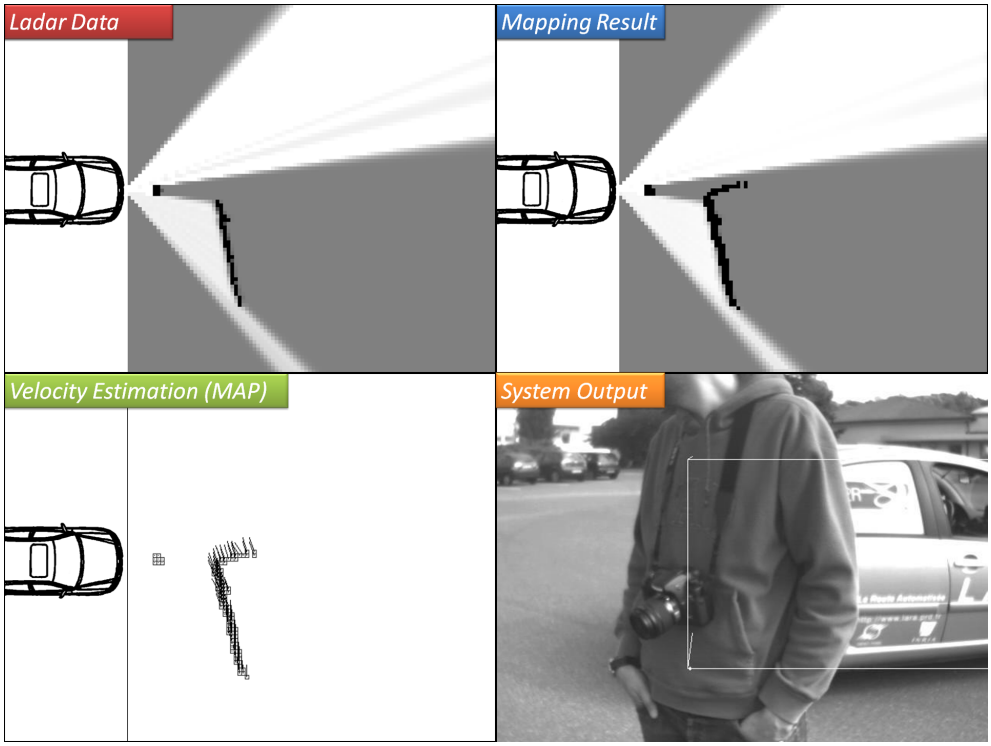
6.8.3 Enhanced Tracking

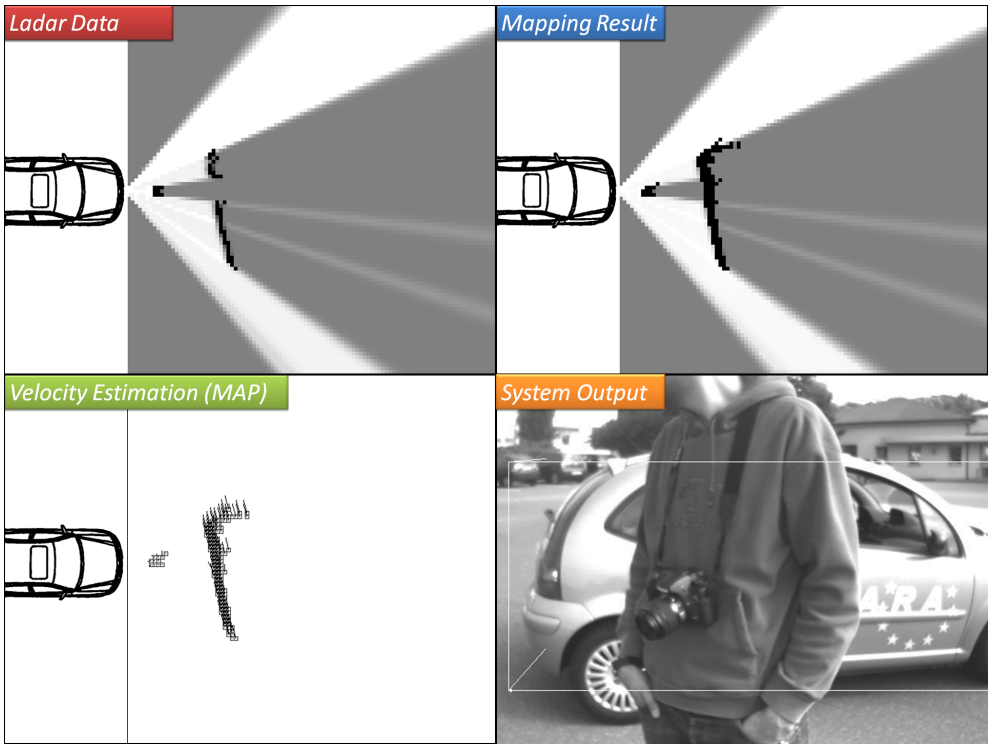
Description

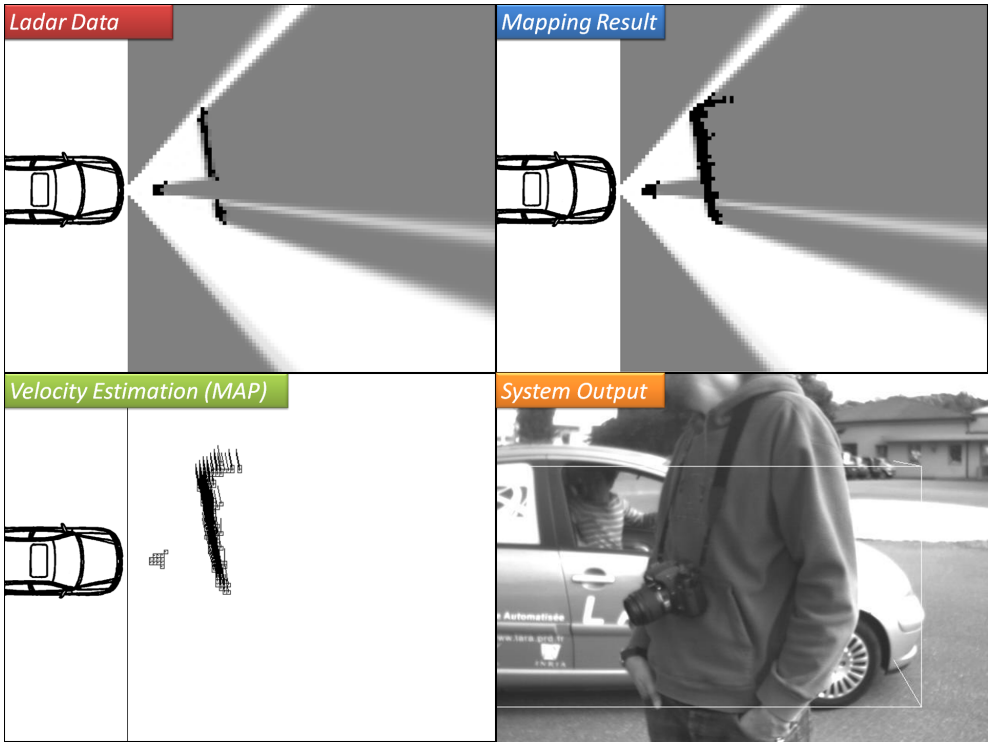
Situations where objects are heavily occluded are also problematic for point-based tracking algorithms. Such a situation is described in Chapter 4 where a portion of a moving vehicle is wrongly classified as a pedestrian by the system of Chapter 3 because of an uncorrect tracking. The same situation processed by the algorithm proposed in this chapter is easily handled as shown in the following figures. Note that the detected car is projected on the camera image for clarity and does not correspond to a misclassified pedestrian.

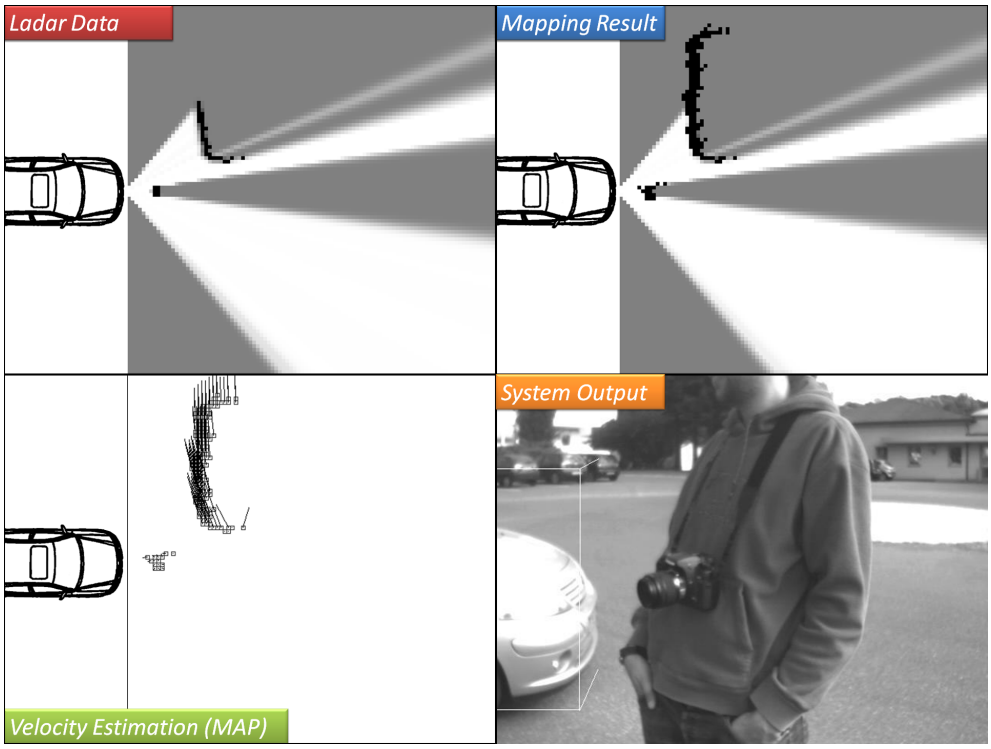
Results




Ladar Data
Mapping Result
Velocity Estimation (MAP)
System Output


Ladar Data
Mapping Result
Velocity Estimation (MAP)
System Output


Ladar Data
Mapping Result
Velocity Estimation (MAP)
System Output


Ladar Data
Mapping Result
System Output
Velocity Estimation (MAP)

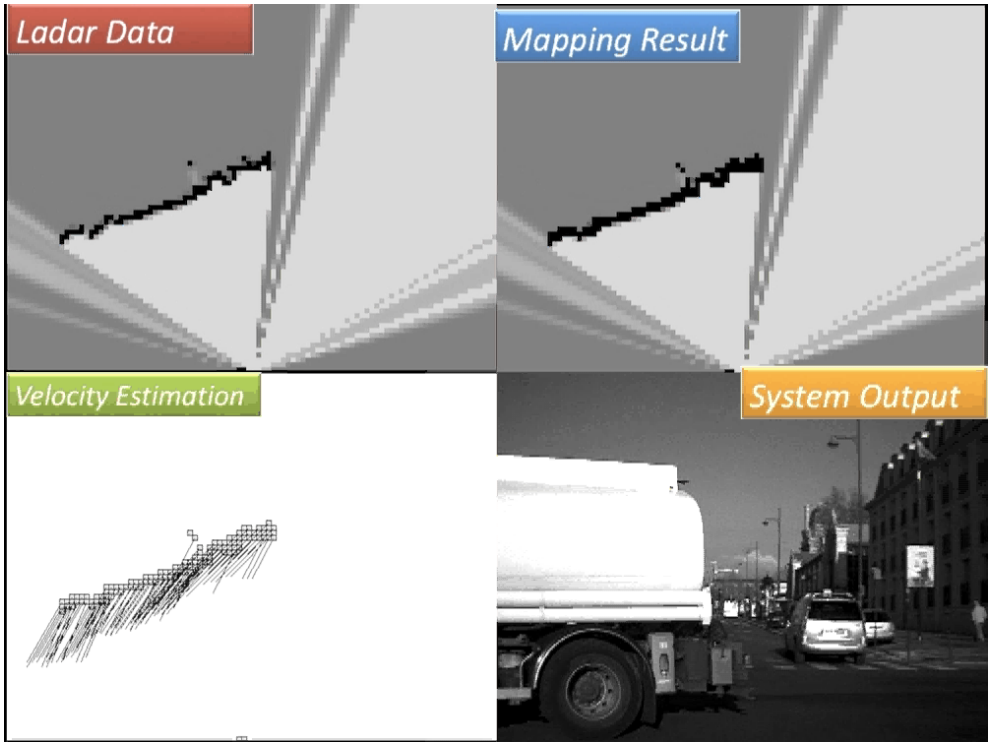
Analysis

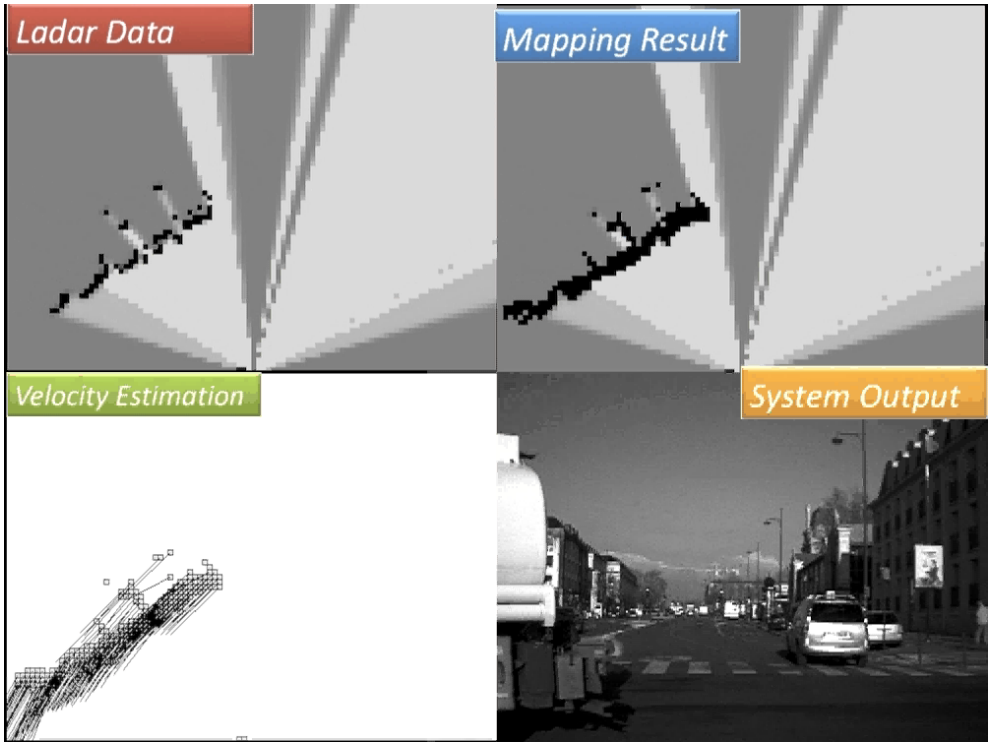
These results show that the proposed approach is able to track objects in situations where point-based tracking algorithms would presumably fail. It is interesting to note that thanks to mapping, the association procedure is not an issue here. Indeed, even when the moving car is partially occluded, the system keeps its outlines in memory and is then naturally able to "understand" why an other object is appearing on the other side of the pedestrian. A great variety of similar situations are perfectly handled by the algorithm proposed in this chapter while leading to significant failures when processed by the algorithm presented in Chapter 3.

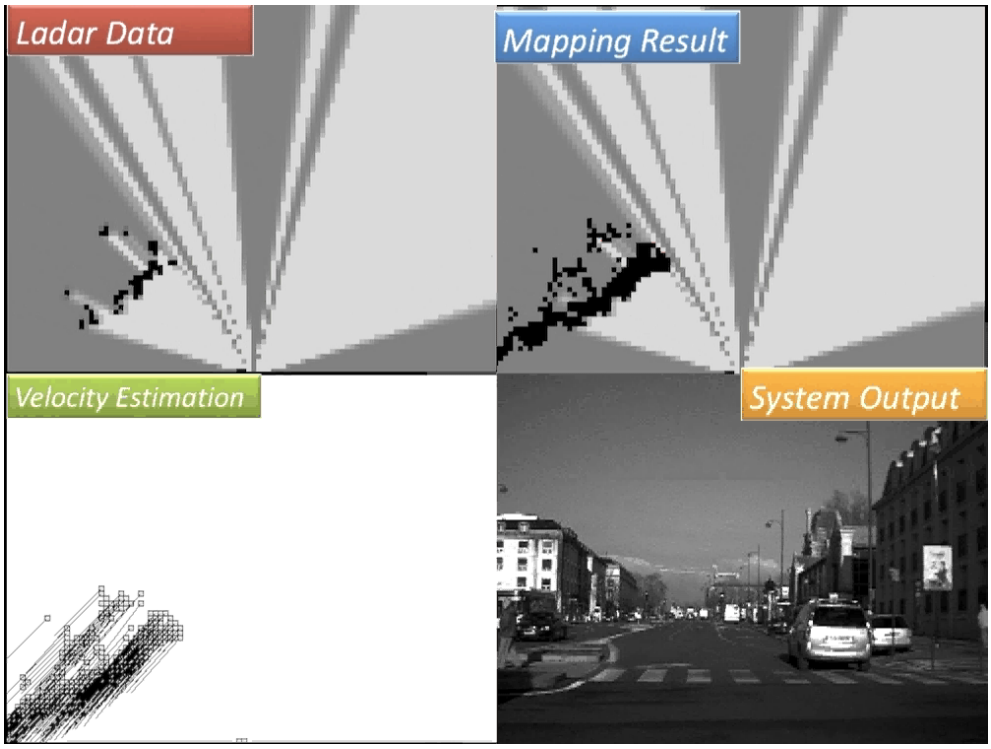
6.8.4 Enhanced Classification

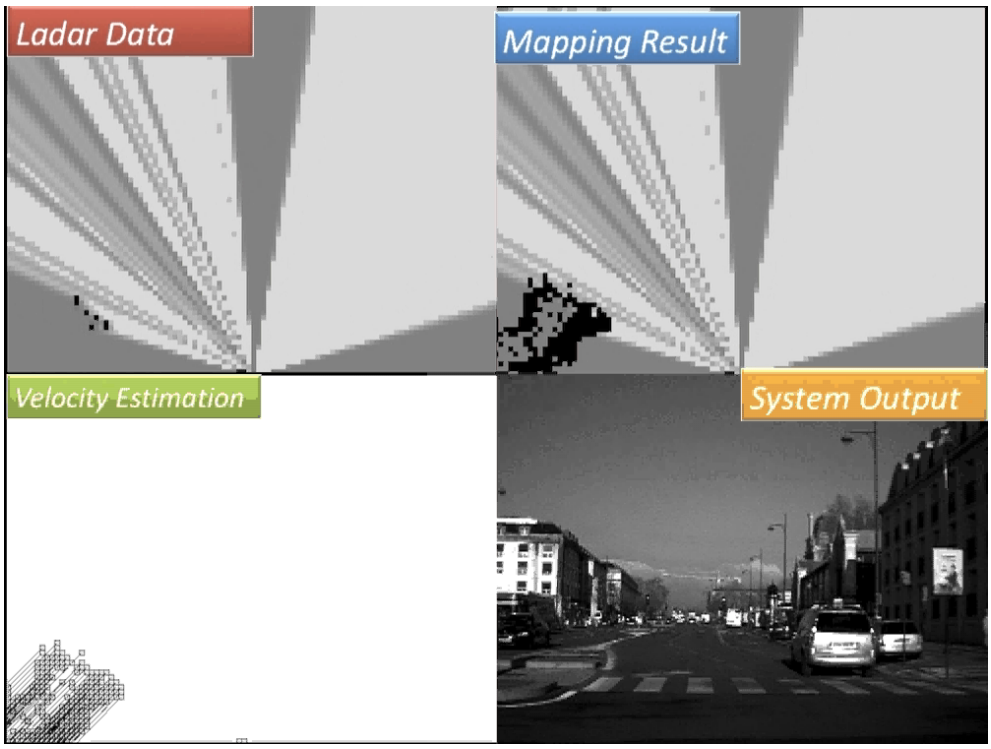
Finally, an important classification issue due to "stair effects" was described in Chapter 4. Objects being seen on various shapes along time are indeed usually difficult to accurately classify. Besides these objects are usually badly structured making any model based approach impossible. By refining the outlines of all objects in the environment, the algorithm proposed in this chapter is able for example to handle this complex situation where a truck can only be accurately detected and classified by aggregating sensor observations over time.

Results - Example 1




Ladar Data
Mapping Result
Velocity Estimation
System Output


Ladar Data
Mapping Result
Velocity Estimation
System Output

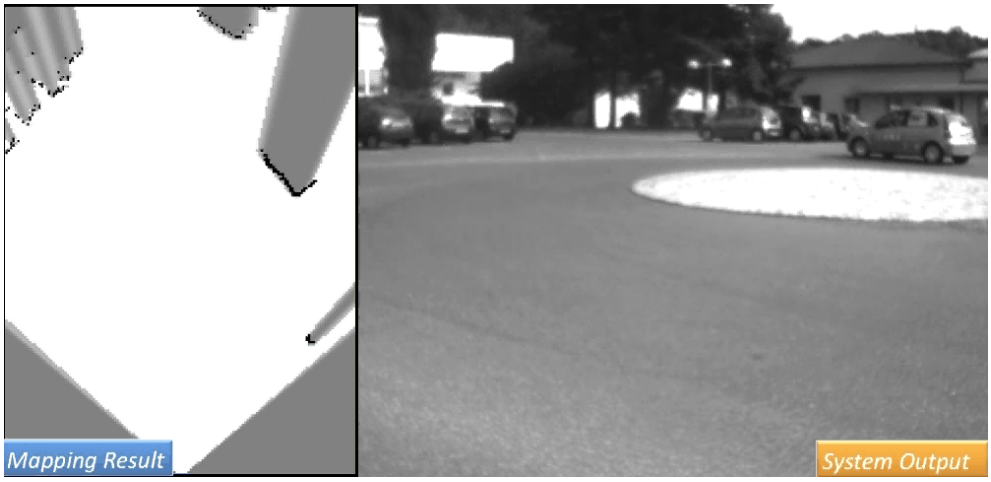


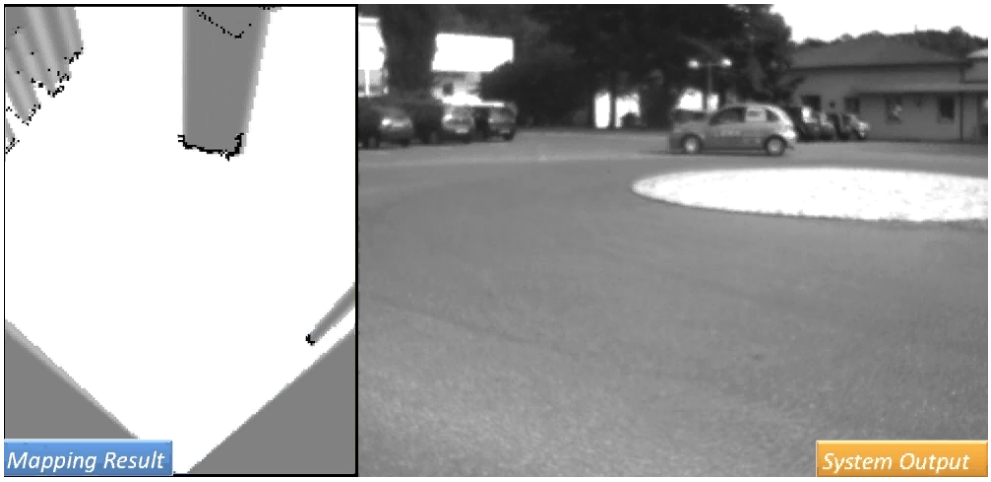
Analysis - Example 1

These results illustrate the ability of the proposed algorithm to cope with badly structured objects. Indeed, the truck shown in this situation is not seen by the sensor as a usual rectilinear object. Because laser impacts are probably hitting a great variety of different surfaces on the truck frame, this moving object is difficult to classify by only looking at successive ladar scans. The original capability of the system to map moving objects is here particularly useful. The consistent estimated map along with the velocity estimates allow to detect the truck as one single object and to provide rich information to a potential truck classifier.

Results - Example 2

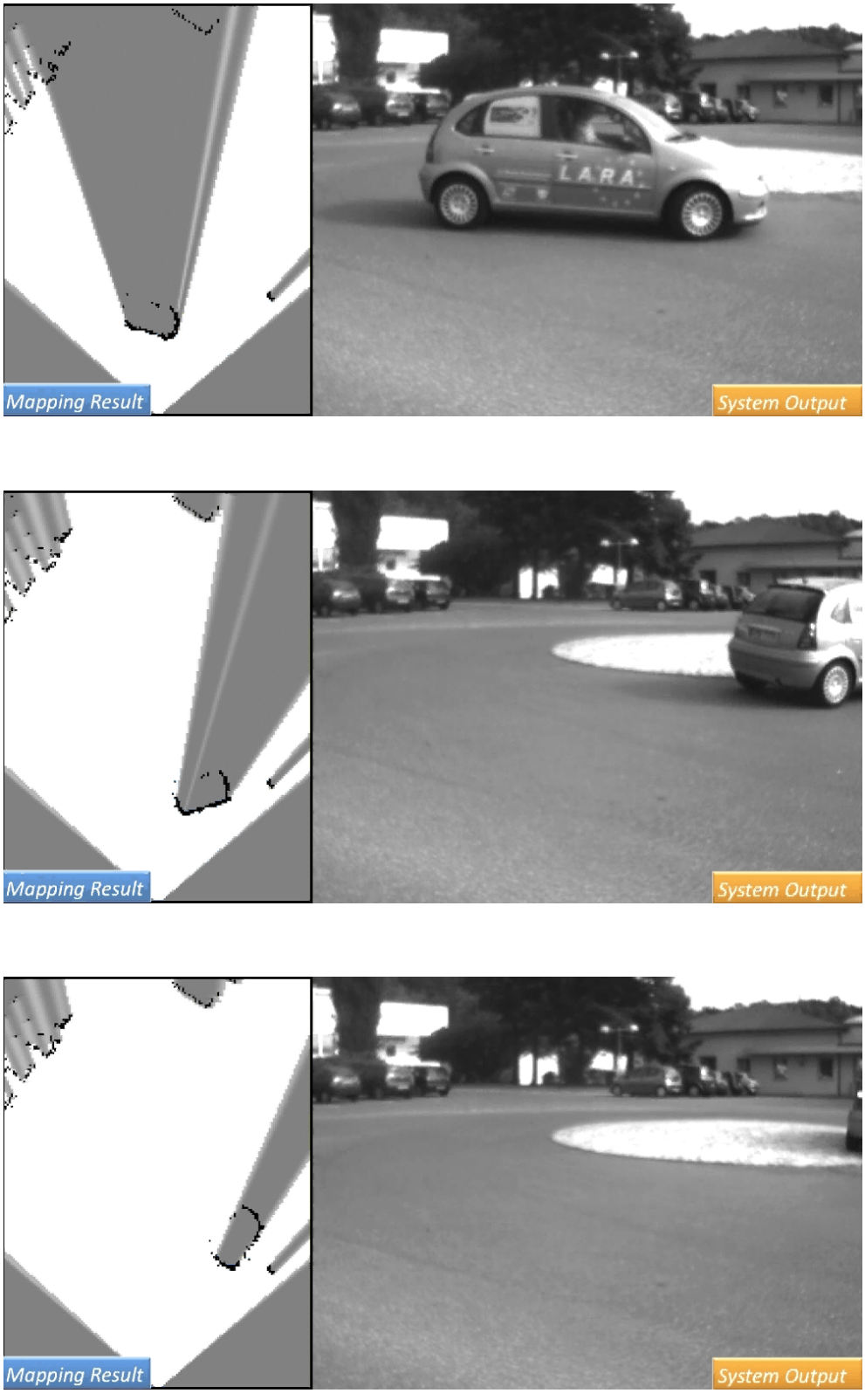
This second situation is not related to the "stair effect" problem but shows another type of situations where the original capabilities of the proposed system can facilitate classification. In this scenario, a car is turning in front of the sensing vehicle. From a ladar point of view, this car will alternatively be seen as a single line or a L-shape object. The mapping capabilities of the proposed approach allow for the exact outlines of the car to be refined over time as seen in the figures below.


Mapping Result
System Output


Mapping Result
System Output


Mapping Result
System Output


Mapping Result
System Output
Mapping Result
System Output
Mapping Result
System Output


Mapping Result
System Output
Mapping Result
System Output
Mapping Result
System Output

Analysis - Example 2

No classification failures are at stake here, but we believe that the ability of the proposed system to construct over time the exact outlines of all the objects in the scene (including moving ones) can facilitate significantly the classification of objets that are well tracked and detected but whose outlines are only periodically visible (which is the case of this car here).

6.9 Conclusion

This chapter presents a pedestrian perception system based on the grid-based approach proposed in Chapter 5. This algorithm is specifically designed to implement three important interactions between the perceptual tasks:

1. Detection $\Longleftrightarrow$ Tracking
2. Tracking $\Longleftrightarrow$ Mapping
3. Classification $\Longleftrightarrow$ Mapping

These interactions have indeed been identified as critical in Chapter 4 to overcome the main fundamental limitations of perception systems based on existing approaches (like the system proposed in Chapter 3). Experiments based on both simulated data and real data show that all the limitations discussed in chapter 4 are simultaneously solved by the proposed algorithm.

First, Tracking is made before Detection allowing robust Detection in situations where the objects are at the same time close from each others and heavily occluded.

Second, Tracking benefits from the algorithm ability to build consistent maps for every object in the environment. This enables the system to handle situations where point-based tracking algorithms would normally fail.

Third, building over time the map of every object in the environment is also an efficient way to solve the problems related to the "stair effect" mentioned in Chapter 4 and to reach better classification performance.

Finally, the proposed algorithm solves the Mapping, Detection, Tracking and Classification tasks in the same mathematical framework. This allows to naturally exploit interactions between perceptual tasks through summation over probability mass functions and to maintain over time a significant level of uncertainty modelling.

While still being more computationally demanding than most existing approaches, the system proposed in this chapter is an original solution to the main problems implied by highly changing environment perception.

Chapter 7

Conclusion

7.1 Summary

This dissertation has two main objectives. First, this work is an attempt to provide a comprehensive analysis about what makes most existing perception systems so sensitive to complex situations. The idea that this was mainly due to under exploited interactions between the perceptual tasks has progressively gain some popularity in the recent years. This led to very interesting and insightful approaches that were mentioned a great number of times in this dissertation.

However, we believe that these recent works referred to as *SLAM with DATMO* approaches only address a tiny part of a bigger problem. By directly implementing the interaction between the *SLAM* module and the *DATMO* module, these approaches are indeed able to cope reasonably well with dynamic environments. But because the level of implemented interactions is still very low, most of these approaches are unable to handle the difficult situations detailed in Chapter 4.

One recent work is however pushing the level of interactivity to the next level. Indeed, Vu and his co-authors have proposed an efficient algorithm that enables the interaction between Detection and Tracking. This simple interaction allows to solve at least two critical issues encountered by most existing approaches. Both detection and tracking are then enhanced provided that the objects of interest can be easily matched with basic primitives.

The work presented in this dissertation can be seen as a natural extension of this movement toward fully interacting perceptual systems. Based on the observation made in Chapter 2 that all the possible interactions between Detection, Tracking, Classification, Mapping and Localization can be mutually beneficial, we proposed in Chapter 5 a unified grid-based framework where all these possible interactions can be elegantly implemented.

Defining such a unified mathematical framework to solve every perceptual task is also a way to ensure that uncertainties can be correctly transmitted from one task to another. As discussed in Chapter 4, modelling and transmitting correctly the uncertainties generated by each algorithm is indeed essential to achieve a high level of reliability.

To prove the validity of this approach, a complete pedestrian system based on the grid-based framework proposed in Chapter 5 is described in Chapter 6. The three critical interactions identified in Chapter 4 have of course been specifically enabled in this system. Experiments conducted on real data show that this algorithm is able, as a consequence, to handle the challenging situations that no other existing approach is capable of managing simultaneously. Unfortunately, this algorithm is also for the moment more computationally demanding than most existing approaches.

7.2 Future extensions

The unified grid-based formalism proposed in this dissertation raises several questions that are not directly addressed in this dissertation. Here are some thoughts about the numerous possible extensions of this work.

Real time Processing

The benefits of the grid-based approach proposed in this dissertation are significant but the implied computational burden is also a significant limitation for onboard implementations. Some interesting methods have been proposed in the last decade to alleviate the computational requirements of grid-based mapping algorithms by using multi-resolution grids. It would be very interesting to adapt these technics to the grid-based general approach proposed in this dissertation and to investigate the possibility to decrease the execution time from one order of magnitude.

Fusion Capabilities

The algorithm derived in Chapter 6 is explicitly designed to process ladar data but the general framework proposed might also be a powerful formalism to fuse heterogeneous data coming from different types of sensors. Data collected by a calibrated camera could for example be directly exploited inside this framework. Color data could for example be used to assist the detection computation, being directly integrated as a measurement likelihood. Velocity information provided by radars could also be integrated in the computation of the velocity estimation and detection problem.

Integration of Higher Level Tasks

The grid-based general approach offers a powerful mathematical paradigm to naturally express the five perceptual tasks as probability mass functions *pmf* estimation problems. While the perception problem is well addressed through these five tasks, higher level tasks related to scene understanding issues could also be implemented within the same framework. For example an additional *pmf* could be introduced to model the location where a point occupying a cell is heading to, at different point in time, in the immediate future or the probability that in a particular cell a collision happen in the next 2 seconds... While not being directly a perception task, the computation of this task would certainly be mutually beneficial with some of them.

7.3 Conclusion

We hope that this work will prove that enabling rich interactions (in terms of uncertainty management) between the five perceptual tasks is an efficient and arguably the only possible way to design perception systems that are truly reliable. It is also our hope that the proposed grid-based framework serves as an insightful basis for further work on the perception of highly changing environments and on the wider problem of autonomous scene understanding.

List of Publications

1. Gwennaël Gâté, Fawzi Nashashibi. *An Approach for Robust Mapping, Detection, Tracking and Classification in Dynamic Environments.* In Proceedings of the IEEE International Conference on Advanced Robotics (ICAR), 2009.

2. Gwennaël Gâté, Amaury Breheret, Fawzi Nashashibi. *Centralised Fusion for Fast People Detection in Dense Environments.* In Proceedings of the IEEE International Conference on Robotics Automation (ICRA), 2009.

3. Gwennaël Gâté, Fawzi Nashashibi. *Fast Algorithm for People and Groups of People Detection and Tracking using a Laser Scanner.* In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2009.

4. Gwennaël Gâté, Amaury Breheret, Fawzi Nashashibi. *Fast Pedestrian Detection in Dense Environment with a Laser Scanner and a Camera.* In Proceedings of the IEEE Vehicular Technology Conference (VTC), 2009.

5. Gwennaël Gâté, Fawzi Nashashibi. *Using Targets Appearance to improve Pedestrian Classification with a Laser Scanner.* In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), 2008.

References

Abramson, Yotam, Steux, Bruno, & Ghorayeb, Hicham. 2007. Yet Even Faster (YEF) real-time object detection. *International Journal of Intelligent Systems Technologies and Applications*, **2**(2/3), 102–112. 71

Andrieu, Christophe, Davy, Manuel, & Doucet, Arnaud. 2003. Efficient Particle Filtering for Jump Markov Systems. Application to Time-Varying Autoregressions. *IEEE Transactions on Signal Processing.* 31

Arras, K.O., Mozos, O.M., & Burgard, W. 2007. Using Boosted Features for the Detection of People in 2D Range Data. *Pages 3402–3407 of: Robotics and Automation, 2007 IEEE International Conference on.* 47

Arulampalam, Sanjeev, Maskell, Simon, Gordon, Neil, & Clapp, Tim. 2001. A Tutorial on Particle Filters for On-line Non-linear/Non-Gaussian Bayesian Tracking. *IEEE Transactions on Signal Processing*, **50**, 174188. 30

Bailey, Tim. 2002. *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments.* Ph.D. thesis, University of Sidney. 27

Bar-Shalom, Y., & Fortman, T. E. 1987. *Tracking and data association.* Academic Press Professional, Inc. 31

Benenson, Rodrigo. 2008. *Perception for driverless vehicle: design and implementation.* Ph.D. thesis, Mines ParisTech. 29

Bertozzi, M., Broggi, A., Chausse, F., Chausse, F., Fascioli, A., & Tibaldi, A. 2003. Shape-Based Pedestrian Detection and Localization. *Page 328333 of: Proceedings of the IEEE International Conference on Intelligent Transportation Systems.* 47

Biber, Peter. 2003. The Normal Distribution Transform: A New Approach to Laser Scan Matching. *In: Proceedings of the IEEE/RJS International Conference on Intelligent Robots and Systems.* 27

Blackman, S. S. 2004. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE*, **19**(1), 518. 32

Blackman, Samuel, & Popoli, Robert. 1999. *Design and Analysis of Modern Tracking Systems.* Artech House Publishers. 23, 32

Blom, Henk A, & Bar-Shalom, Yaakov. 1988. The Interacting Multiple Model Algorithm for Systems with Markovian Switching Coefficients. *Automatic Control, IEEE Transactions on.* 31

Brown, Robert Grover, & Hwang, Patrick Y.C. 1997. *Introduction to Random Signals and Applied Kalman Filtering.* 3rd revised edition edn. 23

Chen, Y., & Medioni, G. 1991. Object modeling by registration of multiple range images. *Page 27242729 vol.3 of: Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on.* 26

Cui, Jinshi, Zha, Hongbin, Zhao, Huijing, & Shibasaki, Ryosuke. 2007. Laser-based detection and tracking of multiple people in crowds. *Computer Vision and Image Understanding*, **106**(2-3), 300–312. 46

Dempster, Arthur. 2008. A Generalization of Bayesian Inference. *Pages 73–104 of: Classic Works of the Dempster-Shafer Theory of Belief Functions.* 73

Dubois, Didier. 1985. *Theorie des possibilites.* 73

Elfes, A. 1989a. Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer*, June, 22(6): 46–57. 26

Elfes, Alberto. 1989b. *Occupancy grids: a probabilistic framework for robot perception and navigation.* Ph.D. thesis, Carnegie Mellon University. 24, 104, 140

Enzweiler, M., & Gavrila, D.M. 2009. Monocular Pedestrian Detection: Survey and Experiments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **31**(12), 2179–2195. 71

Fayad, F., & Cherfaoui, V. 2007. Tracking objects using a laser scanner in driving situation based on modeling target shape. *Pages 44–49 of: Proceedings of the IEEE Intelligent Vehicles Symposium.* 33, 34, 46, 79

Fayad, F., Cherfaoui, V., & Dherbomez, G. 2008. Updating confidence indicators in a multi-sensor pedestrian tracking system. *Pages 156–161 of: Intelligent Vehicles Symposium, 2008 IEEE.* 34

Freund, Y, & Schapire, R. 1999. A short introduction to boosting. *Japonese Society for Artificial Intelligence*, **14**(5), 780, 771. 70

Freund, Yoav, & Schapire, Robert. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. *Pages 37, 23 of: European Conference on Computational Learning Theory.* 70

Fuerstenberg, K., Linzmeier, Dirk T, & Dietmayer, Klaus C. J. 2003. Pedestrian recognition and tracking of vehicles using a vehicle based multilayer laserscanner. *In: Proceedings of the 10th World Congress on Intelligent Transport Systems.* 46

Fuerstenberg, K.Ch., Dietmayer, K.C.J., & Willhoeft, V. 2002. Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner. *Pages 31–35 vol.1 of: Proceedings of the IEEE Intelligent Vehicle Symposium*, vol. 1. 46

Gate, G., & Nashashibi, F. 2009. Fast Algorithm for Pedestrian and Group of Pedestrian Detection using a Laser Scanner. *In: Proceedings of the IEEE Intelligent Vehicles Symposium (IV 2009).* 33

Gate, G., Breheret, A., & Nashashibi, F. 2009. Centralized Fusion for Fast People Detection in Dense Environments. *In: IEEE International Conference on Robotics & Automation (ICRA 2009).* 34

Gidel, S., Checchin, P., Blanc, C., Chateau, T., & Trassoudaine, L. 2008. Parzen method for fusion of laserscanner data: Application to pedestrian detection. *Pages 319–324 of: Proceedings of the IEEE Intelligent Vehicles Symposium (IV 2008).* 33

Gutmann, J.-S., & Schlegel, C. 1996. AMOS: comparison of scan matching approaches for self-localization in indoor environments. *Advanced Mobile Robots, Euromicro Workshop on*, **0**, 61. 26

Hahnel, Dirk, Schulz, Dirk, & Burgard, Wolfram. 2003. Mobile Robot Mapping in Populated Environments. *Advanced Robotics*, **17**, 2003. 38

Herman, Shawn Michael. 2002. *A Particle Filtering Approach To Joint Passive Radar Tracking And Target Classification.* Ph.D. thesis, University of Illinois. 30

Kalman, Rudolph Emil. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASMEJournal of Basic Engineering*, **82**(Series D), 3545. 30

Kraetzschmar, Gerhard K, Gassull, Guillem Pags, Uhl, Klaus, Pags, Guillem, & Uhl, Gassull Klaus. 2004. Probabilistic Quadtrees for Variable-Resolution Mapping of Large Environments. *In: Proceedings of the 5th IFAC/EURON.* 26, 104, 157

Leal, Jeff. 2003. *Stochastic Environment Representation.* Ph.D. thesis, University of Sidney. 104, 117, 121

Mendes, A., Bento, L.C., & Nunes, U. 2004. Multi-target detection and tracking with a laser scanner. *Pages 796–801 of: Intelligent Vehicles Symposium, 2004 IEEE.* 46, 53

Montemerlo, M., & Thrun, S. 2004. A multi-resolution pyramid for outdoor robot terrain perception. *In: Proceedings of the AAAI National Conference on Artificial Intelligence.* 26, 104

Montesano, Luis, Minguez, Javier, & Montano, Luis. 2005. Modeling the Static and the Dynamic Parts of the Environment to Improve Sensor-based Navigation. *In: In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA).* 38

Oh, Songhwai, Russell, Stuart, & Sastry, Shankar. 2004. Markov Chain Monte Carlo Data Association for General Multiple-Target Tracking Problems. *In: Proceedings of the IEEE Conference on Decision and Control.* 32

Papoulis, Athanasios, & Pillai, S.Unnikrishna. 2002. *Probability, Random Variables and Stochastic Processes with Errata Sheet.* 4th edn. McGraw Hill Higher Education. 24

Petrovskaya, A., & Thrun, S. 2008. Model based vehicle tracking for autonomous driving in urban environments. *In: Robotics: Science and Systems IV.* 33, 91

Petrovskaya, Anna, & Thrun, Sebastian. 2009. Model based vehicle detection and tracking for autonomous urban driving. *Auton. Robots*, **26**(2-3), 123139. 34

Prassler, E., Scholz, J., & Fiorini, P. 1999. Navigating a Robotic Wheelchair in a Railway Station during Rush Hour. *The International Journal of Robotics Research*, **18**(7), 711–727. 38

Premebida, C., & Nunes, U. 2006. A Multi-Target Tracking and GMM-Classifier for Intelligent Vehicles. *Pages 313–318 of: Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE.* 46

Reid. 1978. An algorithm for tracking multiple targets. *Pages 1202–1211 of: Decision and Control including the 17th Symposium on Adaptive Processes, 1978 IEEE Conference on*, vol. 17. 32

Schulz, Dirk, Burgard, Wolfram, Fox, Dieter, & Cremers, Armin B. 2001. Tracking Multiple Moving Targets with a Mobile Robot Using Particle Filters and Statistical Data Association. *In: Proceedings of the IEEE International Conference on Robotics and Automation.* 31, 33

Shafer, Glenn. 1976. *Mathematical Theory of Evidence.* Princeton University Press. 73

Shao, Zhao, Huijing, Nakamura, K., Katabira, K., Shibasaki, R., & Nakagawa, Y. 2007. Detection and tracking of multiple pedestrians by using laser range scanners. *Pages 2174–2179 of: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on.* 46

Song, Xuefeng, & Nevatia, Ram. 2005. A Model-Based Vehicle Segmentation Method for Tracking. *Pages 1124–1131 of: Computer Vision, IEEE International Conference on*, vol. 2. 32

Spinello, L., Triebel, R., & Siegwart, R. 2008. Multimodal detection and tracking of pedestrians in urban environments with explicit ground plane extraction. *Pages 1823–1829 of: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on.* 34, 47

Stanciulescu, B., Breheret, A., & Moutarde, F. 2007. Introducing New AdaBoost Features for Real-Time Vehicle Detection. *In: Proceedings of Cognitive Systems with Interactive Sensors.* 71

Streller, D., Furstenberg, K., & Dietmayer, K. 2002. Vehicle and object models for robust tracking in traffic scenes using laser range images. *In: Proceedings of the IEEE International Conference on Intelligent Transportation Systems.* 46

Thrun, Sebastian. 2002. Robotic mapping: A survey. *Exploring Artificial Intelligence in the New Millenium.* 35

Tugnait, Jitendra K. 1981. Detection and estimation for abruptly changing systems. *Pages 1357–1362 of: Proceedings of the 20th IEEE Conference on Decision and Control including the Symposium on Adaptive Processes*, vol. 20. 31

Viola, P., & Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. *Pages I–511–I–518 vol.1 of: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. 70

Viola, P., Jones, M.J., & Snow, D. 2003. Detecting pedestrians using patterns of motion and appearance. *Pages 734–741 vol.2 of: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on.* 70

Vu, Trung-Dung, & Aycard, Olivier. 2009. Laser-based Detection and Tracking Moving Objects using Data-Driven Markov Chain Monte Carlo. *In: IEEE International Conference on Robotics & Automation (ICRA 2009).* 29, 33, 39, 91, 93, 95, 96

Wang, Chieh-Chih. 2004. *Simultaneous Localization, Mapping and Moving Object Tracking.* Ph.D. thesis, Carnegie Mellon University. 31, 38

Wang, Chieh-Chih, & Thorpe, C. 2004. A hierarchical object based representation for simultaneous localization and mapping. 27, 28

Wang, Chieh-Chih, Thorpe, Charles, Thrun, Sebastian, Hebert, Martial, & Durrant-Whyte, Hugh. 2007. Simultaneous Localization, Mapping and Moving Object Tracking. *The International Journal of Robotics Research*, Sept., 889–916. 29, 33, 39, 93, 96

Wender, S., Schoenherr, M., Kaempchen, N., & Dietmayer, K. 2005. Classification of laserscanner measurements at intersection scenarios with automatic parameter optimization. 34, 47

Wolf, Denis F., & Sukhatme, Gaurav S. 2005. Mobile Robot Simultaneous Localization and Mapping in Dynamic Environments. *Autonomous Robots*, **19**(1), 53–65. 33

Xavier, J., Pacheco, M., Castro, D., Ruano, A., & Nunes, U. 2005. Fast Line, Arc/Circle and Leg Detection from Laser Scan Data in a Player Driver. *Pages 3930–3935 of: Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on.* 46

Zhao, H., & Shibasaki, R. 2005. A novel system for tracking pedestrians using multiple single-row laser-range scanners. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, **35**(2), 283–291. 46, 90

Zhao, H., Shao, X.W., Katabira, K., & Shibasaki, R. 2006. Joint Tracking and Classification of Moving Objects at Intersection Using a Single-Row Laser Range Scanner. *Pages 287–294 of: Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE.* 46

Zhao, Tao, Nevatia, Ram, & Wu, Bo. 2008. Segmentation and Tracking of Multiple Humans in Crowded Environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **30**(7), 1198–1211. 32

Zivkovic, Z., & Krose, B. 2007. Part based people detection using 2D range data and images. *Pages 214–219 of: Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on.* 47